ASYMPTOTICS AND INTERPRETABILITY OF DECISION TREES AND DECISION TREE ENSEMBLES

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Yichen Zhou

May 2019

ASYMPTOTICS AND INTERPRETABILITY OF DECISION TREES AND DECISION TREE ENSEMBLES

Yichen Zhou, Ph.D.

Cornell University 2019

Decision trees and decision tree ensembles are widely used nonparametric statistical models. A decision tree is a binary tree that recursively segments the covariate space along the coordinate directions to create hyper rectangles as basic prediction units for fitting constant values within each of them. A decision tree ensemble combines multiple decision trees, either in parallel or in sequence, in order to increase model flexibility and accuracy, as well as to reduce prediction variance. Despite the fact that tree models have been extensively used in practice, results on their asymptotic behaviors are scarce. In this thesis we present our analyses on tree asymptotics in the perspectives of tree terminal nodes, tree ensembles and models incorporating tree ensembles respectively. Our study introduces a few new tree related learning frameworks for which we can provide provable statistical guarantees and interpretations.

Our study on the Gini index used in the greedy tree building algorithm reveals its limiting distribution, leading to the development of a test of better splitting that helps to measure the uncertain optimality of a decision tree split. This test is combined with the concept of decision tree distillation, which implements a decision tree to mimic the behavior of a block box model, to generate stable interpretations by guaranteeing a unique distillation tree structure as long as there are sufficiently many random sample points.

Meanwhile, we apply mild modification and regularization to the standard tree boost-

ing to create a new boosting framework named Boulevard. The major difference Boulevard has in contrast to the original framework is our integration of two new mechanisms: honest trees, which isolate the tree terminal values from the tree structure, and adaptive shrinkage, which scales the boosting history to create an equally weighted ensemble. With carefully chosen rates, we establish consistency and asymptotic normality for Boulevard predictions. This theoretical development provides us with the prerequisite for the practice of statistical inference with boosted trees.

Lastly, we investigate the feasibility of incorporating existing semi-parametric models with tree boosting. We study the varying coefficient modeling framework with boosted trees applied as its nonparametric effect modifiers, because it is the generalization of several popular learning models including partially linear regression and functional trees. We demonstrate that the new framework is not only theoretically sound as it achieves consistency, but also empirically intelligible as it is capable of producing comprehensible model structures and intuitive visualization.

BIOGRAPHICAL SKETCH

Yichen Zhou was born in Nanjing, China where he spent his childhood with his parents and grandmother. After attending Nanjing Foreign Language School (which doubled the coursework of English learning) for six years and graduating in summer 2009, he got early admitted to Tsinghua University, Beijing due to his satisfying performance in contests of mathematics and informatics, and left his hometown in the following fall.

During undergraduate he majored in pure and applied mathematics in the Department of Mathematical Sciences where he learned the jargons and humors that could only be appreciated by mathematicians. His narration and practice of mathematics, including greek letter calligraphy, was profoundly influenced by Dr. Xuguang Lu. In 2013, he received his Bachelor in Science in mathematics with a thesis on camera calibration advised by Dr. Binheng Song, and graduated as an Excellent Graduate.

After spending another year working with Dr. Ke Deng on Bayesian statistics in the Yau Mathematical Sciences Center, he joined the PhD program in statistics at Cornell University and moved to Ithaca, New York in Fall 2014. His general interests were at the boundary between statistics and machine learning, or in other words, statistical learning theories. His research focusing on the statistical guarantees of tree models was advised by Dr. Giles Hooker. During his five year stay in Ithaca, Yichen also cultivated the professionalism of being a statistician through teaching, consulting and industry internships. He defended his work in May 2019.

Dedicated to my parents, Ping Zhou and Shuqin Ding.

ACKNOWLEDGEMENTS

I find myself a lucky person having encountered and known so many great people during my PhD study who have given me endless support and help. This piece of work would never be possible without them.

First and for most, I am extremely honored and grateful to have Giles Hooker, Thorsten Joachims, and Sumanta Basu on my special committee guiding my PhD study along the way. In particular, as my advisor, Giles deserves special credits. I could still remember my first meeting with him during which he humbly self-claimed to be a non-mainstream statistician, only after which did I get to gradually be astounded by his broad knowledge, deep insights, generosity of the time with students and responsible advisoring. In my mind Giles is the standard for a good statistician and a good mentor. I would also like to express my gratitude to the department faculty members for their creating an inclusive working environment through all the lectures, seminars and lab meetings.

There are also other people in the department from whom I have also get tremendous help: Beatrix, Diana and Phillip. Thank you all for helping keep our timetable running smoothly.

Outside of Cornell, I owe my gratitude to Daria Sorokina, François Huet and Vamsi Salaka, three brilliant mentors I met during my summer internships. Their industrial vision influenced my view of good statistics from the perspective of real world applications.

Apart from work, I also feel honored to have had an awesome statistics PhD cohort. People from my year: Daniel, Ben, Skyler, Wenyu, Wentian and Xiaoyun, I appreciate and cherish the nights we spent together on homeworks (and boardgames). As a person whose genotype is “uncommon among elite strength athletes”, I would also express my gratitude

to my friends who keep me in a fit form: Yang, for your crossfit training (scaled) and mountaineering education; Skyler, for being my gym buddy when we needed to make 50 visits every semester; Zhengze, for being my another gym buddy who is also the mentee of Yang; and Chang, for your beginner level squash instruction.

Most importantly, there are people in my life who have shaped me to be who I am, and no words are enough to deliver my sincere appreciation. My parents, Ping Zhou and Shuqin Ding, my grandparents Xiufang Tang, Xijia Ding and Yuezhen Zhang and my other family members. You have amazing visions. All the teachers who changed my view of the world: Lianhua Liu, for introducing me to music. Jianhua Xing, for getting me into mathematics and computer science. Köhler Shi and Xiaoqin Yao, for introducing me to English and sciences. And all my friends that are too many to name.

TABLE OF CONTENTS

Biographical Sketch . . . iii
Dedication . . . iv
Acknowledgements . . . v
Table of Contents . . . vii
List of Tables . . . x
List of Figures . . . xi

1 Decision Trees and Decision Tree Ensembles 1
1.1 Statistical Learning . . . 1
1.2 Decision Trees . . . 2
1.3 Decision Tree Ensembles . . . 7
1.4 Outlines . . . 8

2 Distillation Trees and Their Stability Measure 11
2.1 Interpreting Black Boxes . . . 11
2.1.1 Gini Indices . . . 13
2.2 A Test of Better Split . . . 14
2.2.1 Asymptotic Distribution of Gini Indices . . . 15
2.2.2 Comparing Two Splits . . . 18
2.2.3 Sequential Testing . . . 19
2.2.4 Multiple Testing . . . 20
2.3 Stable Distillation . . . 21
2.3.1 Choice of Prospective Splits . . . 22
2.3.2 Generating Points . . . 23
2.3.3 Stopping Rules . . . 24
2.4 Empirical Study . . . 26
2.4.1 Simulated Data . . . 26
2.4.2 Real Datasets . . . 31
2.4.3 Binary Classification . . . 32
2.4.4 Multiclass Classification . . . 34
2.4.5 Stability . . . 36
2.5 Model Fitting v.s. Distillation . . . 39

3 Boulevard Boosted Trees and Their Asymptotics 41
3.1 Gradient Boosted Decision Trees and Boulevard . . . 41
3.1.1 Boulevard . . . 45
3.2 Honest Trees and Forests . . . 46

3.2.1 Honest Trees and Honest Forests . . . 46
3.2.2 Adaptivity of Boosted Trees . . . 50
3.3 Boulevard Convergence . . . 52
3.3.1 Stochastic Contraction and Boulevard Convergence . . . 52
3.3.2 Beyond L^2 Loss . . . 56
3.4 Asymptotic Normality . . . 56
3.4.1 Building Deeper Trees . . . 57
3.4.2 Fixed Design . . . 58
3.4.3 Missing Terminal Subsample . . . 59
3.4.4 Exponential Decay of Influence and Asymptotic Normality . . . 60
3.4.5 Random Design . . . 63
3.4.6 Undersmoothing, Tree Space Capacity and Subsampling . . . 67
3.5 Eventual Non-adaptivity . . . 68
3.5.1 Local Homogeneity and Contraction Regions . . . 70
3.5.2 Escaping the Contraction Region . . . 71
3.6 Empirical Study . . . 73
3.6.1 Predictive Accuracy . . . 74
3.6.2 Limiting Distribution . . . 75
3.6.3 Reproduction Interval . . . 77
3.7 Proofs . . . 79
3.7.1 Properties of Tree Structure Matrices . . . 80
3.7.2 Stochastic Contraction . . . 81
3.7.3 Asymptotic Normality . . . 85

4 Tree Boosted Varying Coefficient Models and Their Asymptotics 91
4.1 Combining Parametric Models with Boosting . . . 91
4.1.1 Models under VCM . . . 94
4.1.2 Trees and VCM . . . 95
4.2 Tree Boosted Varying Coefficient Models . . . 97
4.2.1 Notations . . . 97
4.2.2 Boosting Framework . . . 97
4.2.3 Local Gradient Descent with Tree Kernels . . . 100
4.2.4 Examples . . . 102
4.3 Asymptotics . . . 104
4.3.1 Tree Boosted VCM with L^2 Loss . . . 105
4.3.2 Decomposing Decision Trees . . . 106
4.3.3 Consistency . . . 108
4.4 Empirical Study . . . 112

4.4.1 Identifying Signals . . . 112
4.4.2 Model Accuracy . . . 113
4.4.3 Visual Interpretability: Beijing Housing Price . . . 114
4.4.4 Fitting Other Model Class . . . 116
4.5 Shrinkage, Selection and Serialization . . . 117
4.6 Proofs . . . 121

5 Discussion and Potential Future Work 130
5.1 U-statistics and Boosting . . . 130
5.2 Stochastic Contraction, Shrinkage, Dropout and Second Order Method . . 131
5.3 Partially Linear Model Inference . . . 134
5.4 Varying Coefficient Models, Functional Trees and Tree Distillation . . . 135
5.5 Model Extrapolation and Manipulation . . . 136

LIST OF TABLES

2.1 Dataset description showing the number of covariates, the number of training points, the number of testing points and the levels of responses for each dataset. 31

2.2 Stability of BASE and AppTree. The table shows the number of identical structures out of 100 replications and counts the occurrences of the top 3 structures in each case. Cnt for counts. Boldfaced numbers show the occurrences of the dominant tree structure out of 100 replications generated by AppTree for each dataset. 38

3.1 Prediction standard deviations scale with error standard deviations. . . . 78

4.1 Prediction accuracy of classification and 0-1 loss for six UCI data sets through tenfold cross validation. Results are shown as mean(sd). Sources of some datasets are: BANK(Moro et al., 2014) and OCCUPANCY(Candanedo and Feldheim, 2016). 114

4.2 Prediction accuracy of regression and mean square error for six UCI data sets through tenfold cross validation. Results are shown as mean(sd). Sources of some datasets are: BEIJINGPM(Liang et al., 2015), BIKEHOUR(Fanaee-T and Gama, 2014), ONLINENEWS(Fernandes et al., 2015) and ENERGY(Tsanas and Xifara, 2012). 115

4.3 Fitting a partially linear model using tree boosted VCM. Plot on the left shows the nonparametric intercept. Table on the right shows the coefficients of predictive covariates. 117

LIST OF FIGURES

1.1 A classification tree predicting labels on a two dimensional covariate space. LHS is the spatial segmentation, and RHS the visualized tree. . . 4

2.1 Predictive accuracy of RF, CART and AppTree. Results of RF and CART are recalculated for but are theoretically not affected by different values of N_{ps}. 28
2.2 Mimicking accuracy. PROB compares RF and AppTree by the L^1 difference of their class probabilities. CLASS compares by the predicted class labels. 28
2.3 Stability of AppTree with different Nps values. The top 4 layers and top 5 layers of the trees are summarized respectively. In each column, a single black bar represents a unique structure of the tree, while the height of the bar represents the number of occurrence of that structure out of 100 replications. 29
2.4 Performance evaluation on binary classification datasets. From top to bottom: CAD-MDD, BreastCancer, Car, ClimateModel. From left to right: ROC curves, consistency with RF on testing set, consistency with RF on new data points. 33
2.5 Performance evaluation on multiclass (3-class) classification datasets. ROC curves are plotted in a one v.s. all fashion. Consistency is only checked on testing data. From top to bottom: Cardiotocography, WineRed, WineWhite. 35
2.6 BASE and AppTree stability measured on binary classification datasets. From left to right: CAD-MDD, BreastCancer, Car, ClimateModel. In each column, a single black bar represents a unique structure of the tree, while the height of the bar represents the number of occurrence of that structure out of 100 replications. 36
2.7 BASE and AppTree stability measured on multiclass classification datasets. From left to right: Cardiotocography, WineRed, WineWhite, Abalone. In each column, a single black bar represents a unique structure of the tree, while the height of the bar represents the number of occurrence of that structure out of 100 replications. 37

3.1 Training and testing error curves of tree ensembles on silmulated data. . 75
3.2 Training and testing error curves of tree ensembles on real world data sets. 76

3.3 Distributions of predictions of test points with different error terms. The errors are N(0,1), Unif[-1,1], equal point mass at $\{-1, 1\}$, and half chance -1 half chance Unif[0,2], respectively. 77
3.4 Reproduction intervals. Boxplots show distributions of predictions; red intervals are reproduction intervals; blue dots are truths. Sample sizes are 1000 (top row) and 5000 (bottom row), error terms Unif[-1,1] (left column) and Unif[-2, 2] (right column). Coverage is shown by numbers next to interval centers. 79

4.1 Example of varying coefficient mappings on the action space under the OLS settings. 103
4.2 Example of varying coefficient mappings on the action space under the logistic regression settings. 104
4.3 Histograms of distributions of fitted coefficient values. Color code: ground truth (grey) and tree boosted VCM (red). 113
4.4 Beijing housing unit price broken down on several factors. 115

CHAPTER 1

DECISION TREES AND DECISION TREE ENSEMBLES

1.1 Statistical Learning

One can image countless situations in our everyday life when we have to make a decision out of a few observations. To give a handful of examples: A psychiatrist may need to diagnose a patient with depression based on their response on a mental health questionnaire. A dealer may want to price a used car based on its make, model, year, milage and history report. An outdoor person may intend to choose a location for their weekend hike based on local temperature, chance of precipitation, trail length and trail difficulty. And a hungry graduate student may have to decide where to grab a meal based on the type of food, average waiting time, price and whether the place is open at a certain time.

Most, if not all, of these situations can be described mathematically under the settings of statistical classification and regression. We have some observations X which are called covariates, predictors or features depending on the subject area, that can be numeric (price of food, milage of car) or categorical (answer to a yes-no question, type of food) and a response Y between which we want to summarize and justify certain relationship f connecting them so that

$$Y \approx f(X).$$

When we have a sample consisting of a sizable number of X's and Y's, the predictive perspective of statistics helps to determine f, and the inferential perspective of statistics

helps to study the properties of f. In particular, we call these analyses *classification* when Y is a categorical label, and *regression* when Y is a numeric value. We call f *parametric* if the shape of f is determined by a fixed number of parameters independent of the sample size, and *nonparametric* otherwise. In spite of various terminologies involved, the shared crucial idea behind all is that we want to apply a statistical methodology to study the relationship between the covariates and the responses.

This data driven procedure has a modern name called *learning* partially due to the explosion of data and the practice of using machines (computers) intensively to perform the underlying modeling. Since we will mostly focus on the statistical perspectives of this procedure, we will refer to it as *statistical learning* in this thesis.

1.2 Decision Trees

Classification and Regression Trees (CART), or Decision Trees, have become a popular branch and an iconic choice of nonparametric statistical learning methods since its first introduction by Breiman et al. (1984). As per its original design, a decision tree is a binary tree splitting the covariate space along the coordinate directions to create hyper rectangles called terminal nodes. Each split in the tree is determined by a greedy strategy to best distinguish the observations in its two branches respectively, or in other words, to minimize a given impurity measure. After that, each terminal node in the tree is assigned a fitted value that is usually decided by the average or the majority vote. Figure 1.1 is an example of a classification tree, and the following algorithm demonstrates the generic

steps of constructing a decision tree.

Algorithm 1.1 (CART)**.**

- *Start at the root node corresponding to the full covariate space.*
- *Given a node, enumerate all possible candidate splits by going over all covariates and collecting all possible split values.*
- *Choose a split yielding the maximal impurity reduction based on an impurity measure to separate the node, thus the corresponding covariate subspace, into two child branches.*
- *Work recursively in the child node to further split until a stopping criterion is met.*
- *Prune the tree.*
- *Calculate the fitted value in each of the terminal nodes.*

Besides the actual greedy building algorithm, there are multiple alternative perspectives to view decision trees.

- A decision tree is a piecewise constant estimate of the underlying relation between covariates and responses. It is the finite linear combination of hyper rectangular indicator functions. This point of view allows a potentially deep decision tree to reach any given level of accuracy thanks to the Littlewood's three principles stating any Lebesgue measurable function can be approximated by a finite sum of scaled interval indicators to any required precision.

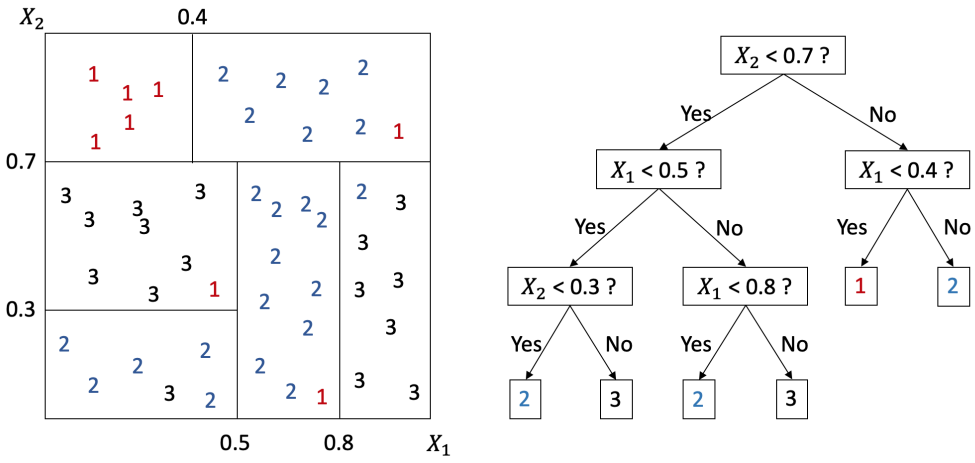


Figure 1.1: A classification tree predicting labels on a two dimensional covariate space. LHS is the spatial segmentation, and RHS the visualized tree.

- A decision tree is an adaptive nearest neighbor smoother where the adaptive distance measure between two points is given by the likelihood of them being in the same terminal node compared to other methods relying on a metric on the covariate space. In other words, a decision tree is capable of creating a topology on the covariate space adaptively describing the similarity between observations should we keep expanding the tree.
- A decision tree is the exclusive and exhaustive combination of binary decision paths mimicking human decision making, where a decision rule consists of evaluating dichotomously a few if-then predicates. Meanwhile, the states of all predicates summarize all possible results of all decision rules.

Due to these advantages, decision trees can achieve decent empirical performance

without relying on structural assumptions regarding the underlying space. However, they remain mathematical intractable as the result of several characteristics.

- Decision trees are nonparametric. Their behavior cannot be described by using a few parameters.
- Decision trees are step functions, therefore they are neither smooth, Lipshitz nor even continuous. Methods including the use of differentiation or with the underlying assumption of continuity, for instance the attempt to Taylor expand a decision tree, are not directly compatible with trees.
- Decision trees are adaptive to the training sample through their greedy building algorithm, which forces the analyses to condition on the sample. This behavior may challenge us to discover a feasible mathematical formulation for the greedy algorithm, as well as create potential difficulties when we intend to isolate the tree from the data to reach unconditioned conclusions.
- The space of all decision trees within a certain depth is not closed under addition. Moreover, the completion of the space of all decision trees contains all measurable functions. In other words, the model space is oversized, and for meaningful subspaces there is no proper low dimensional description, for instance basis expansion, to serve as an analyzable mathematical simplification.

Endeavors have been made to modify the building algorithm in an attempt to enable and simplify the analysis of decision trees. Most of these modifications target the mechanism used for deciding where to place the splits.

- Completely randomized trees or uniform trees (see Biau et al., 2008; Biau, 2012; Scornet et al., 2015; Scornet, 2016) that construct trees while ignoring the impurity measure. They place random split locations in the trees, and retrospectively decide the fitted values in the terminal nodes using the sample.
- Semi-randomized trees (see Wager and Walther, 2015; Wager and Athey, 2017) require that each covariate has a minimal chance of being selected as the splitting covariate for any split in the tree. This guarantee is achieved by a partially random split rule.
- Dyadic trees (see Blanchard et al., 2004, 2007) only evaluate splits at the midpoints of each of the intervals of possible covariates.
- Two sample trees (honest trees) (see Wager and Athey, 2017) utilize another independent sample to decide the tree structure in the CART manner, then decide terminal values with the actual sample.

The common idea behind the aforementioned methods is the partial separation of the training sample and the greedy algorithm so that the worst case behavior of the resulting tree can be controlled by the tree structures untethered from the training sample. These modifications help to develop the asymptotic properties of decision trees while preserving most of their tree characteristics. However, the cost we pay for this theoretical soundness is the empirical practicality and the intuitive comprehensibility.

1.3 Decision Tree Ensembles

Creating an ensemble of weak learners is an effective practice to scale up model complexity, improve accuracy and reduce variance, leading us to tree ensembles when the involved weak learners are decision trees. Popular tree ensembles, which are listed below, differ in the ways of how many trees there are in the ensemble, how much randomness they cast into training each component tree, and how much dependence each tree is allowed with the rest of the ensemble.

- Bagging, short for bootstrap aggregating (Breiman, 1996; Bühlmann et al., 2002). Bagging creates a tree ensemble of any size by training each tree on a randomly selected subsample and averaging all trees.
- Random forests (Breiman, 2001). Random forests are similar to bagging with the difference that each split covariate in any component tree is now chosen in a randomly selected subset of covariates as well.
- Boosting (Friedman, 2001). Gradient boosted decision trees create a sequential ensemble of trees during whose construction the last tree is fitted on the mismatch described by the functional gradient of its current status of the tree ensemble.
- Additive groves (Sorokina et al., 2008), which fix the number of trees and construct the ensemble by extending the depth of its component trees through backfitting.
- Bayesian additive regression trees (Chipman et al., 2010), which also create an ensemble of a given size. They are similar to additive groves in the sense that back-

fitting is used to update the ensemble, while each update selects the structure of the tree based on a Bayesian prior on all tree shapes.

Among all mentioned methods, bagged trees and random forests maintain a certain level of conditional independence among their component trees, therefore are easier to analyze. On the other hand, the mathematical formulation of boosted trees belongs to the domain of time inhomogeneous Markov processes which, despite being more efficient in practice, possess a sequentially dependent structure that changes along the boosting history. This fact, along with the ambiguity induced by the greedy tree building strategy, introduces more difficulties to the analyses of boosted ensembles.

1.4 Outlines

In brief, we would like to try answering the following three main questions in this thesis.

1. How should we construct decision trees to assure their stability or honesty?
2. How can we guarantee the asymptotic behaviors of tree boosting?
3. How can we properly interpret the results produced by tree models?

We will separate our answers and further discussions in the following chapters.

There is a decent amount of recent literature discussing the potential approaches to study tree ensembles with alternative tree building strategies and carefully chosen rates.

Stochastic inequalities are introduced to establish concentration bounds for trees without thoroughly investigating their structures. For random forests, the U-statistics framework is effective and has produced substantial results on the asymptotic normality of random forest predictions and as a consequence, a few variance estimators.

For tree boosting we anticipate two plausible solutions both of which suggest us to formulate better mathematical profiles for boosted trees. One is through modifying and regularizing the behavior of boosting for the purpose of obtaining a mathematically friendly form, which is mostly done by adaptively weighting the component trees in the ensemble. The other one is through empirical process theories that relate boosting to its population version process. This approach is more dense in mathematics but has more flexibility as long as the population counterpart of boosting generates tractable mathematical objects.

On interpretability, we would like to treat the interpretability of a statistical learning model from two angels. One is the statistical interpretability, meaning the extent to which we can guarantee the behavior of the model and perform sophisticated statistical inferences. The other is the perceptional interpretability, in other words, model transparency and feasibility that we can point to and explain the exact actions undertaken by the model. Decision trees are inherently intelligible models that can be utilized to perform model distillation for complex black boxes. They align well with our understanding of perceptional interpretability as a universal tool to reason for decision making processes. We will discuss a few practices to make them more effective. Meanwhile, decision tree ensembles are black boxes with few statistical tools to estimate and infer their behavior. Our plans for them will concentrate more on the statistical interpretability in order to develop a set

of conclusions with which we get more knowledge of their doings and perform statistical inferences accordingly.

CHAPTER 2

DISTILLATION TREES AND THEIR STABILITY MEASURE

2.1 Interpreting Black Boxes

Random forests (RF) (Breiman, 2001) and other statistical learning methods have been widely used across different disciplines and are acknowledged for their outstanding predictive power (Caruana and Niculescu-Mizil, 2006). However, statistical learning models may suffer from a trade-off between predictive accuracy and model interpretability (Breiman et al., 2001; Friedman et al., 2001). Black box models, to which we refer as models with complex inherent structures that are relatively impenetrable by standard mathematical analyses, are usually found to be capable of achieving high predictive accuracy due to their fitting power and flexibility. A modern example of black boxes is the deep neural network, which has been extensively used in areas of image, sound and natural language processing while its inner working is still hard to explain and tune. On the other hand we have the concept of glass boxes which are models transparent for inspection. With the presence of both model calsses, in Domingos (1997) the author introduced and experimented the concept of Combined Multiple Models(CMM) that learns a glass box from a black box. Its modern revision can be approached by developing intelligible *student* models which mimic the predictions of the original *teacher* black box: a strategy encompassed by the term *model distillation*. Within model distillation, common student models are generalized additive models (GAMS: see Lou et al. (2012); Tan et al. (2017), Hooker (2007) provides a link between these and PDPs) and decision trees (Breiman et al.,

1984; Quinlan, 1987), which are our focus. Similar work can be found in Johansson et al. (2011, 2010) where the authors discussed the concept of coaching a decision tree by a complex model. He et al. (2012) showed such procedure has desirable theoretical and empirical performance.

Decision trees are attractive as a statistical learning technique. However, the greedy algorithm used to build trees results in high variability and poor performance when used directly on training data. This is because small perturbations of the data used to build the tree can result in dramatically different models as when, for example, a different covariate is chosen in a high-level split with consequences that cascade through the rest of the tree structure. In the context of model distillation, this instability is an important concern: an explanation or interpretation of a learning outcome that is sensitive to small changes in the data may be viewed as unreliable.

In order to obtain a stabilized structure for a decision tree, we take advantage of our ability to generate an arbitrarily large data set from the teacher model. Specifically, we follow Gibbons et al. (2013) in generating pseudo data from a kernel density estimate based on the observed covariates and using the value of the teacher model at these points as a response. In this chapter we additionally ensure that, were this pseudo data to be re-generated, the same tree structure would be chosen with high probability. To carry this out, at each node we assess the stability of the selected split via a hypothesis testing framework; when splitting, we generate a large enough corpus of pseudo data to ensure that separation between the Gini index split criterion at the chosen split and that of other candidates is large enough to be consistently selected. This framework is repeated at each

split to obtain a stabilized tree, generating new pseudo data as needed.

As our experiments show, this can result in the need to generate very large sets of pseudo-data when competing splits produce very similar improvement and achieving a stabilized tree can be computationally demanding. We think that this is an important observation: that many existing uses of decision trees in model distillation may produce unstable model interpretations or explanations and our understanding of these models may rest more on the particular data used to generate the approximation tree than on the underlying structure of the teacher. There are some subtle distinctions to be made here: if a distillation tree replaces the learned model when making predictions, we might reasonably choose to present it as an explanation for how a prediction is made, even if the structure of the tree was originally determined partially by chance. However, if we also hope to interpret reasoning behind the prediction, or expect the tree to explain something about the teacher, we would require explanations to be reproducible.

2.1.1 Gini Indices

Most tree building procedures, i.e. C4.5 (Quinlan, 2014), select splits based on maximizing the information gain (minimizing the impurity) that results from each candidate split point. There are multiple choices of defining the information gain in the literature (Breiman et al., 1984; Loh and Shih, 1997). In this chapter, we will focus on the Gini information associated with the Gini index as its empirical estimator. For the distribution

of (X, Y) where $Y \in \{1, \ldots, k\}$ the category labels of X, one way to write the Gini gain g is

$$g = \sum_{i \neq j} P(Y = i)P(Y = j) = 1 - \sum_{i=1}^{k} P(Y = i)^2. \tag{2.1}$$

The empirical version of this formula defines the corresponding Gini index. It is worth noticing that this conventional definition implies more information with a smaller value, meaning smaller Gini indices implies less discrepancy among responses. The formula also indicates its relation with the sample variance.

When splitting a node in the decision tree, we divide the sample space into two subsets within each of which the responses are more uniform than in the whole space, increasing the total information gain. This value is estimated by the weighted sum of the two Gini indices after splitting, hence the split with the maximal Gini index implies the best information gain and is therefore employed. In the following sections we will show that, in our approximation setting, we can determine our sample size to get more precise estimate of the Gini indices, thereby stabilizes the split at each node.

2.2 A Test of Better Split

In this section, we will develop a means of assessing the stability of a node splitting procedure via the use of hypothesis tests. This will then be employed to ensure that we generate enough data to reliably choose the same split points.

Consider a multiclass classification problem. The original sample consist of covariates and responses $\{(\tilde{X}_i, \tilde{Y}_i)\}_{i=1}^{n_0}$ where $\tilde{X}_i \in \mathbb{R}^m$, $\tilde{Y}_i \in \{1, 2, \ldots, k\}$, m the dimension of covariate

space, and k the levels of responses. We obtain a black box classifier $\mathcal{F}$ from the sample. $\mathcal{F}$ will later serve as the oracle we try to mimic, generating points (pseudo sample) $\{(X_i, Y_i)\}_{i=1}^n$ of arbitrary size n. Here $X_i = (X_i^1, \ldots, X_i^m) \in \mathbb{R}^m$, and $Y_i = (Y_i^1, \ldots, Y_i^k) \in \mathbb{R}^k$ are the $\mathcal{F}$-predicted class probabilities over responses. To approximate $\mathcal{F}$, our tree classifiers will be constructed from $\{(X_i, Y_i)\}_{i=1}^n$.

We now wish to control the probability that two different pseudo sample points, $\{(X_i, Y_i)\}_{i=1}^n$ and $\{(X_i^*, Y_i^*)\}_{i=1}^n$ would result in different splits. Here, we make pairwise comparisons between the current best split, and the list of candidate alternatives. For each alternative, the p-value for a test that the difference in Gini gains is greater than zero gives us an estimate of the probability that a different dataset would choose the alternative over the current best split. Summing these probabilities gives a bound on the likelihood of splitting the current node a different way and we then select n to control this probability.

2.2.1 Asymptotic Distribution of Gini Indices

A theoretical discussion of the evaluation of splits can be found in Banerjee et al. (2007). In our specific case, we compare the Gini indices of candidate splits: To do so, we examine their asymptotic behavior and obtain a central limit theorem (CLT) so normal based tests can be developed. (2.1) implies an averaging over sample when calculating the Gini index, suggesting the existence of this CLT.

To examine two perspective splits G_1 and G_2 with the same sample, their Gini gains

are

$$g_1 = 1 - \pi_{1,l}\left(\sum_{j=1}^{k} \theta_{1,l,j}^2\right) - \pi_{1,r}\left(\sum_{j=1}^{k} \theta_{1,r,j}^2\right),$$
$$g_2 = 1 - \pi_{2,l}\left(\sum_{j=1}^{k} \theta_{2,l,j}^2\right) - \pi_{2,r}\left(\sum_{j=1}^{k} \theta_{2,r,j}^2\right),$$

where π represents the covariate distribution of $\tilde{X}$ and θ the conditional probability of $\tilde{Y}$ given $\tilde{X}$. Subscripts are arranged in the order of the split (1 for G_1 and 2 for G_2), the left (denoted as l) or right (denoted as r) child, and the class label from 1 to k. For instance,

$$\pi_{1,l} = P(G_1(X) = 0), \quad \pi_{1,r} = P(G_1(X) = 1), \quad \theta_{1,l,j} = P(Y = 1 | G_1(X) = j),$$

and respectively for G_2. The empirical versions, Gini indices, are

$$\hat{g}_{1,n} = 1 - \frac{n_{1,l}}{n}\sum_{j=1}^{k}\left(\hat{\theta}_{1,l,j}\right)^2 - \frac{n_{1,r}}{n}\sum_{j=1}^{k}\left(\hat{\theta}_{1,r,j}\right)^2,$$
$$\hat{g}_{2,n} = 1 - \frac{n_{2,l}}{n}\sum_{j=1}^{k}\left(\hat{\theta}_{2,l,j}\right)^2 - \frac{n_{2,r}}{n}\sum_{j=1}^{k}\left(\hat{\theta}_{2,r,j}\right)^2.$$

Moving to the two children of both splits, we denote the numbers of sample and the ratios of class labels in each child by, for $p \in \{1, 2\}$, $j \in \{1, \ldots, k\}$,

$$n_{p,l} = \sum_{i=1}^{n} 1_{\{G_p(X_i)=0\}}, \quad \hat{\theta}_{p,l,j} = \frac{1}{n_{p,l}}\sum_{i=1}^{n} Y_i^j \cdot 1_{\{G_p(X_i)=0\}},$$
$$n_{p,r} = \sum_{i=1}^{n} 1_{\{G_p(X_i)=1\}}, \quad \hat{\theta}_{p,r,j} = \frac{1}{n_{p,r}}\sum_{i=1}^{n} Y_i^j \cdot 1_{\{G_p(X_i)=1\}},$$

and create the following stacked vectors to denote the sample version and the population version of the number of pseudo-sample points that should fall in each category as in the

CLT, which is for $p \in \{1,2\}, q \in \{l,r\}$,

$$N_{p,q} = \begin{bmatrix} n_{p,q}\hat{\theta}_{p,q,1} \\ \vdots \\ n_{p,q}\hat{\theta}_{p,q,k} \end{bmatrix}, \quad \Theta_{p,q} = \begin{bmatrix} \pi_{p,q}\theta_{p,q,1} \\ \vdots \\ \pi_{p,q}\theta_{p,q,k} \end{bmatrix}, \quad \sqrt{n}\left(\frac{1}{n}\begin{bmatrix} N_{1,l} \\ N_{1,r} \\ N_{2,l} \\ N_{2,r} \end{bmatrix} - \begin{bmatrix} \Theta_{1,l} \\ \Theta_{1,r} \\ \Theta_{2,l} \\ \Theta_{2,r} \end{bmatrix} \right) \longrightarrow N(0,\Sigma).$$

To relate this limiting distribution to the difference of Gini indices we shall employ the δ-method. Consider the analytic function $f : \mathbb{R}^{4k} \to \mathbb{R}$ s.t.

$$f(x_1,\ldots,x_{4k}) = -\frac{1}{\pi_{1,l}}\sum_{i=1}^{k} x_i^2 - \frac{1}{\pi_{1,r}}\sum_{i=k+1}^{2k} x_i^2 + \frac{1}{\pi_{2,l}}\sum_{i=2k+1}^{3k} x_i^2 + \frac{1}{\pi_{2,r}}\sum_{i=3k+1}^{4k} x_i^2.$$

The δ-method imples that

$$\sqrt{n}\left(f\left(\frac{1}{n}\begin{bmatrix} N_{1,l} \\ N_{1,r} \\ N_{2,l} \\ N_{2,r} \end{bmatrix} \right) - f\left(\begin{bmatrix} \Theta_{1,l} \\ \Theta_{1,r} \\ \Theta_{2,l} \\ \Theta_{2,r} \end{bmatrix} \right) \right) \longrightarrow N(0,\Theta^T\Sigma\Theta), \tag{2.2}$$

where

$$\Theta = f'\left(\begin{bmatrix} \Theta_{1,l} \\ \Theta_{1,r} \\ \Theta_{2,l} \\ \Theta_{2,r} \end{bmatrix} \right) = 2\begin{bmatrix} -\Theta_{1,l} \\ -\Theta_{1,r} \\ \Theta_{2,l} \\ \Theta_{2,r} \end{bmatrix} \in \mathbb{R}^{4k}, \quad \Sigma = cov\begin{bmatrix} N_{1,l} \\ N_{1,r} \\ N_{2,l} \\ N_{2,r} \end{bmatrix} = cov\begin{bmatrix} Y \cdot 1_{\{G_1(X)=0\}} \\ Y \cdot 1_{\{G_1(X)=1\}} \\ Y \cdot 1_{\{G_2(X)=0\}} \\ Y \cdot 1_{\{G_2(X)=1\}} \end{bmatrix} \in \mathbb{R}^{4k\times 4k}.$$

We should point out that while (2.2) provides us with the CLT we need to assess the difference between two Gini indices: expanding (2.2) yields

$$\sqrt{n}\left((\hat{g}_{1,n} - \hat{g}_{2,n}) - (g_1 - g_2)\right) \longrightarrow N(0,\Theta^T\Sigma\Theta).$$

or asymptotically,

$$(\hat{g}_{1,n} - \hat{g}_{2,n}) - (g_1 - g_2) \sim N\left(0, \frac{\Theta^T \Sigma \Theta}{n}\right).$$

Hence, by replacing Θ, Σ by the empirical versions from the pseudo sample, we write

$$\hat{g}_{1,n} - \hat{g}_{2,n} \sim N\left(g_1 - g_2, \frac{\hat{\Theta}^T \hat{\Sigma} \hat{\Theta}}{n}\right). \tag{2.3}$$

2.2.2 Comparing Two Splits

The above formula (2.3) gives rise to the following test when comparing two splits with different batches of pseudo sample. Suppose we have two prospective splits G_1 and G_2. After drawing pseudo sample $\{(X_i, Y_i)\}_{i=1}^n$ and observing without loss of generality that $\hat{\Delta}_n = \hat{g}_{1,n} - \hat{g}_{2,n} < 0$. We intend to claim that G_1 is better than G_2. In order to ensure this split is chosen reliably, we can run a single-sided test to check whether we would obtain the same decision when accessing $\hat{\Delta}_n^* = \hat{g}_{1,n}^* - \hat{g}_{2,n}^* < 0$ with another independently-generated set of pseudo sample $\{(X_i^*, Y_i^*)\}_{i=1}^n$. Assume that $\{(X_i, Y_i)\}_{i=1}^n$ and $\{(X_i^*, Y_i^*)\}_{i=1}^n$ are independent samples, (2.3) implies

$$\hat{\Delta}_n^* - \hat{\Delta}_n \sim N\left(0, \frac{2\hat{\Theta}^T \hat{\Sigma} \hat{\Theta}}{n}\right),$$

which gives,

$$\hat{\Delta}_n^* \Big| \left(\hat{\Delta}_n = \hat{g}_{1,n} - \hat{g}_{2,n}\right) \sim N\left(\hat{g}_{1,n} - \hat{g}_{2,n}, \frac{2\hat{\Theta}^T \hat{\Sigma} \hat{\Theta}}{n}\right).$$

This distribution leads to a prediction interval based on which we would get the prediction of the Gini difference using a different pseudo sample. In order to control $P(\hat{\Delta}_n^* < 0)$ at a

confidence level $1 - \alpha$, we need

$$\hat{g}_{1,n} - \hat{g}_{2,n} < Z_\alpha \cdot \sqrt{\frac{2\hat{\Theta}^T \hat{\Sigma} \hat{\Theta}}{n}}, \tag{2.4}$$

where Z is the inverse c.d.f. of a standard normal. With a sufficiently large n it is possible to always determine the better split between G_1 and G_2 should they have any difference. In addition, by combining this test with a pairwise comparisons procedure, we are capable of finding the best split among multiple prospective splits.

2.2.3 Sequential Testing

The power of this better split test increases with n. Since we need to determine n to reveal any detectable difference between two splits, when no prior knowledge is given regarding the magnitude of the difference, we need an adaptive approach to increasing n accordingly.

For a fixed confidence level α, suppose we have tested at sample size n and get p-value $p_n > \alpha$. Referring to (2.4), we have

$$\sqrt{n} \cdot \frac{\hat{g}_{1,n} - \hat{g}_{2,n}}{\sqrt{2\hat{\Theta}^T \hat{\Sigma} \hat{\Theta}}} = Z_{p_n}.$$

Notice that $\frac{\hat{g}_{1,n} - \hat{g}_{2,n}}{\sqrt{2\hat{\Theta}^T \hat{\Sigma} \hat{\Theta}}}$ is the estimator of $\frac{g_1 - g_2}{\sqrt{2\Theta^T \Sigma \Theta^T}}$ which is an intrinsic constant with respect to the pairwise comparison. Hence in order to reach a p-value less than α we may increase sample size to n' such that

$$\sqrt{n'} \cdot \frac{\hat{g}_{1,n} - \hat{g}_{2,n}}{\sqrt{2\hat{\Theta}^T \hat{\Sigma} \hat{\Theta}}} = Z_\alpha,$$

which yields that

$$\sqrt{\frac{n}{n'}} = \frac{Z_{p_n}}{Z_\alpha}. \tag{2.5}$$

Due to pseudo sample randomness, a few successive increments are required before we land in the confidence level. We also need an upper bound for n' and a default split order in case the difference between two splits is too small to identify.

2.2.4 Multiple Testing

So far we have obtained a method to compare a pair of splits. But when splitting a certain node we usually need to choose the best split among multiple $G_1, \dots G_m$. If we still want to test at a certain significance α whether the split with the lowest estimated Gini index, i.e, $\hat{g}_{n,(1)}$, is the optimal, we can perform multiple pairwise comparison and control the familywise error rate using standard procedure like Bonferroni (Dunnett, 1955) or Benjamini-Hochberg (Hochberg and Benjamini, 1990). For example using Bonferroni, we can

- test the hypotheses $H_{i,0} : g_{(1)} = g_{(i)}, i = 2, \dots, t$. Get the p-values $p_2, \dots, p_t$, and
- use $\sum_{i=2}^{t} p_t$ as the p-value of the multiple comparison.

This test aggregates all significance levels into one, presumably resulting in a conservative estimate as we ignore much of the correlation structure of the splits. In this scenario, the updates of sample size made in sequential testing should also be modified as we are now

taking the aggregated significance level. A quick and feasible fix is to replace the p_n in (2.5) by the aggregated significance level. Alternatively, we may just test between the best two splits.

Because of the computational cost, when we have two splits that cannot be distinguished, the sequential and multiple testing procedure may end up demanding an extremely large number of points to make the test significant. In practice, we halt the testing early at a cutoff of certain amount N_{ps} of points, and choose the current best split. This compensation for computation time might lower the real power of the test, leading to a less stable result.

2.3 Stable Distillation

To build an approximation tree, we replace the greedy splitting criterion by our stabilized version within the CART construction algorithm. At each node, we first generate an initial number of pseudo sample points belonging to this node from the black box. Then we compare prospective splits simultaneously based on this set and decide whether we either choose the one with the smallest Gini index with certain confidence or request more pseudo sample points. In the latter case, we keep generating until the pseudo sample size reaches what is required by the sequential testing procedure. This is repeated until we distinguish the best split. We perform this procedure on any node that needs to split during construction to get the final approximation tree.

Algorithm 2.1 (Black Box Distillation)**.**

- *take input: black box predictor $\mathcal{F}$, covariate distribution of X.*
- *return output: approximating tree $\mathcal{T}$.*
- *for current approximating tree V, we check*
  - *if V satisfies some stopping condition, then return V.*
  - *otherwise, we generate n pseudo sample points from $\mathcal{F}$ and find prospective splits $G_1, \cdots, G_m$.*
  - *we keep running sequential testing and generating more pseudo sample points were we not able to distinguish the best split among $G_1, \cdots, G_m$.*
  - *expand V accordingly.*

There are several parameters to tune for this algorithm. We first need all the parameters for CART, for instance the maximal depth of the tree, or maximal and minimal number sample points in each leaf node. We must also choose α to control the significance of the test of better split, and N_{ps} which controls the maximal amount of pseudo sample points we require at each node.

2.3.1 Choice of Prospective Splits

Most methods of finding prospective splits for a decision tree are compatible with our method once they target at optimazing some information gain (Quinlan, 2014, 1987). In building an approximating tree, we only consider making splits at those points which would have been employed in a tree generated from the original training data. We look at

the original sample points that have been carried along the path and take the possible combinations of the covariates and their middle points of adjacent values that have appeared in those sample points.

The reason for deciding prospective splits on the data rather than the black box itself is due to the fact that the black box does not carry any information regarding the true generative distribution of covariates. We would like to estimate the distribution by the empirical distribution plus some random perturbation in the purpose of learning how the black box extrapolates. We will show this in detail in the following section.

Although this method will initially generate a large number of prospective splits, because of the sequential testing scheme, most of those splits will be identified as far worse than the best after a few tests and can be discarded, leaving a negligible effect on the overall performance. In practice, we implement a scheme (Benjamini and Hochberg, 1995) to adaptively discard splits that perform far worse than the current best. All splits are ordered by their p-values against the current best split, and the splits fall below the threshold are discarded.

2.3.2 Generating Points

To generate the pseudo sample, we first generate pseudo covariates then obtain predictions from the black box to get the responses. It is worth noticing that the first step here may encounter the obstacle that, in practice, we do not have the true generative distribution of covariates.

There are quite a few conventional statistical methods we can implement here. Some methods focus on estimating the underlying distribution by smoothers (Wand and Jones, 1994), while the others use bootstrapping or residual permutation to directly manipulate and reorganize the sample points to generate more sample points. In the purpose of exploring more of the covariate space, we take the first approach and use a Gaussian kernel smoother upon the empirical distribution of the sample points. This translates to generating pseudo covariates from observed covariates plus random noise. In the case of discrete covariates, we choose a neighboring category with a small probability. These steps should be considered as a prerequisite information of our method as its main target is to approximate the empirical distribution, which diverges from our oracle coaching task. Therefore the variance of the random noise and the probability of jumping to a neighboring category should be empirically decided.

When we go further down the approximating tree, the covariate space may be narrowed down by the splits along the path. A feasible covariate generator can thus be produced by only smoothing the empirical distribution of those original sample points that have been carried on by this path. We further check the boundary condition to ensure that the covariates we generated agree within the region divided by the splits along the path.

2.3.3 Stopping Rules

Another crucial point to this algorithm is the stopping rule deciding when there is no need to further split a node. Our test is capable of distinguishing any small gap between two

candidate splits at the presence of a sufficiently large pseudo sample, as a result we can ideally build until each node is constant, which is definitely impractical.

We can consider three approaches. The first one, as well as the most straightforward one, is to keep expanding the tree to a preset depth. The advantage of this fixed depth strategy is mostly on the practical side when we want to have the tree depth to either make sense for the actual application (i.e. the length of a decision making path) or to be able to model interactions to the extent reflecting the number of covariates along the path, which is also the order of interaction.

The second approach is through configuring a threshold so that only when all candidate split pairs have a discrepancy below the threshold do we cease the expansion. The occurrence of such situation is a sign indicating the unnecessariness of further splitting. Notice that it is also directly compatible with our test of better split as to model the split discrepancy. The obstacle we encounter for this method is that we can theoretically exhaust all possible candidate splits, which adds much overhead to the multiple testing and actual computation. The performance will depend on our choice of candidate splits to compare.

Alternatively, we can take a third approach to test the signal level in the current node to decide whether the signal is heterogenous enough to support further splits, or in other words, whether the variability inside the node is purely caused by noise. This method requires a measure of the inner node uncertainty, which can be done using tools of random forest variance estimate (Mentch and Hooker, 2016). The development of this stopping rule requires more mathematical justification and is out of the scope of this thesis.

For our empirical study in the following section we choose the use the straightforward strategy of setting the tree depth *a priori*.

2.4 Empirical Study

In this section we choose random forests (RF) as the black box prediction function, thus in the following context RF and black box may be used interchangably. However, our method and analysis can be easily generalized for other predictors by using their prediction instead. We have conducted empirical studies on both simulated and real data to show how the performance of approximating tree compares with both decision trees and RFs. The performance is mainly assessed in three ways: prediction accuracy, consistency with the RF (mimicking accuracy), and stability.

2.4.1 Simulated Data

We experiment our method on a simple simulated dataset to check its behavior. Assume $\tilde{X} \in \mathbb{R}^5$ and $\tilde{Y} \in \{0, 1\}$, and let the covariate distribution $\tilde{X} = (x_1, \ldots, x_5) \sim \text{Unif}[0, 1]^5$.

Write $p = P(\tilde{Y} = 1|\tilde{X})$ and let

$$\text{logit}(p) = \begin{cases} 2, & x_1 > 0.5, \quad x_2 > 0.7, \\ -3, & x_1 > 0.5, \quad 0.7 \geq x_2 > 0.2, \\ -4, & x_1 > 0.5, \quad x_2 \leq 0.2, \\ 3, & x_1 \leq 0.5, \quad x_5 \leq 0.5, \quad x_3 + x_4^2 \geq 1.4, \\ 2, & x_1 \leq 0.5, \quad x_5 \leq 0.5, \quad 1.4 > x_3 + x_4^2 \geq 0.5, \\ -2, & x_1 \leq 0.5, \quad x_5 \leq 0.5, \quad x_3 + x_4^2 < 0.5, \\ 2, & x_1 \leq 0.5, \quad x_5 > 0.5. \end{cases}$$

The generative distribution is intentionally set to be almost tree-structured so the result should reflect our method working under ideal conditions. We do so to avoid extreme cases during our check, while general distributions will be tested on using real datasets.

We compare across three methods: classification trees (CART), random forests (RF) and our proposed approximating tree (AppTree). During each replication, we generate 1,000 sample points from above distribution and obtain a standard RF consisting of 100 trees and a 5-layer CART tree. Then we build a 5-layer approximating tree via the algorithm above. The significant level α for the test of better split is set to be 0.1, and the maximal number of pseudo sample points at each node N_{ps} is set to be $10^4, 10^5$ and 10^6 respectively. For each N_{ps} we have 100 replications. For assessing stability, we use the same setting above but fix one RF as an oracle and learn it by an approximating tree 100 times with $10^4, 10^5$ and 10^6 respectively.

In order to evaluate predictive accuracy and consistency, we generate new covariates

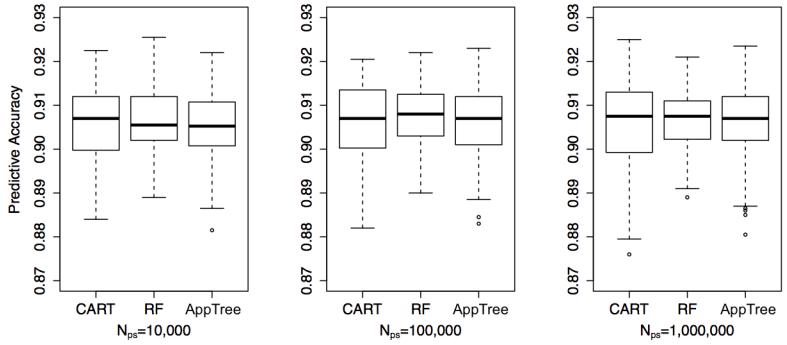


Figure 2.1: Predictive accuracy of RF, CART and AppTree. Results of RF and CART are recalculated for but are theoretically not affected by different values of N_{ps}.

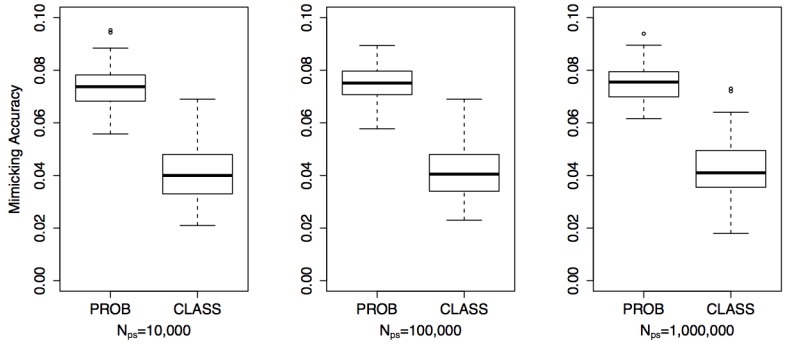


Figure 2.2: Mimicking accuracy. PROB compares RF and AppTree by the L^1 difference of their class probabilities. CLASS compares by the predicted class labels.

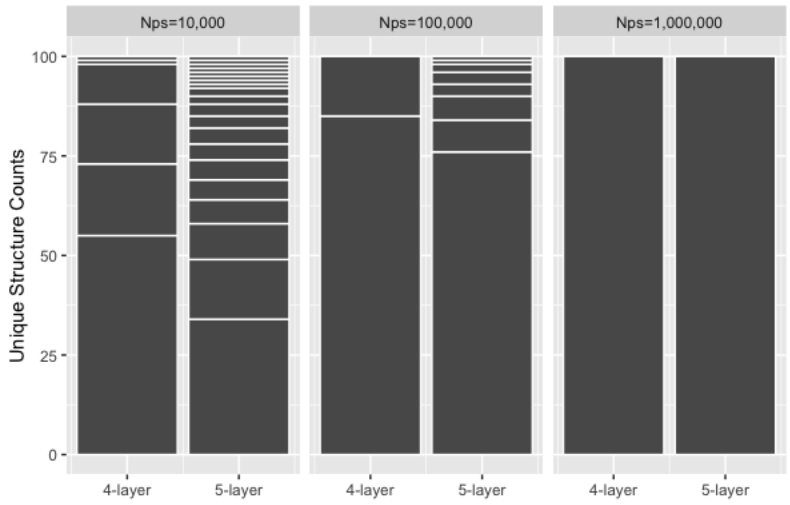


Figure 2.3: Stability of AppTree with different Nps values. The top 4 layers and top 5 layers of the trees are summarized respectively. In each column, a single black bar represents a unique structure of the tree, while the height of the bar represents the number of occurrence of that structure out of 100 replications.

and measure how much the predictions of approximating tree agree with those of the RF. To measure stability, which is defined in our case as the structural uniqueness, we construct multiple approximating trees out of a single RF and look into the variation in their structures. The better split test does not always guarantee a consistent pick through multiple trials due to the pseudo sample randomness, hence we hope to see small variation among all the trees built. We also examine the trees at different depths to capture the variation along the tree growth. In this chapter, we are more interested in the consistency with RF and the stability of the approximating tree. However, we will still compare the predictive accuracy of approximating trees with other models.

Figure 2.1 shows the predictive accuracy of the three methods on new test points. On

average they share similar predictive accuracy; RF has the smallest variance, followed by AppTree. This meets our expectation that AppTree is capable of inheriting stability from the RF after learning how the RF extrapolates. Since the relation between the covariates and the responses is relatively simple, increasing N_{ps} does not significantly improve the performance.

Figure 2.2 shows the comparison between RF and AppTree in terms of the L^1 difference of their predicted class probability, and the disagreement of their class labels. Again the increase of N_{ps} does not bring significant improvement to performance. AppTree has achieved 95% agreement on average with the RF. By expanding the trees to larger sizes the mimicking accuracy can still be marginally increased by "overfitting" the RF.

Figure 2.3 shows the stability of AppTree viewing from its top 4 layers and top 5 layers. It can be seen that by increasing the cap on the maximal number of pseudo sample points AppTree can generate, its stability gets increased significantly. One unique structure is obtained when $N_{ps} = 10^6$, which means that some node actually require $\sim 10^6$ points to detect the best split. Two key observations can be made here. Our control of α is relatively conservative due to our sequential testing and multiple testing steps. The maximal number of pseudo sample needed may be quite large to detect the best split. Overall, this initial check shows results as we expected.

2.4.2 Real Datasets

In this section we will show the results of our method on eight datasets. Seven of them are available on the UCI repository (Lichman, 2013) and one is the CAD-MDD data used by Gibbons et al. (2013). We manually split each dataset into train and test for cross validation. Table 2.1 shows the number of covariates, training and testing sample size and the levels of responses for each dataset.

Name	#Cov	#Train	#Test	Response Levels
CAD-MDD	88	500	336	0,1
BreastCancer(Mangasarian et al., 1995)	30	350	218	0,1
Car	6	1000	727	0,1
ClimateModel(Lucas et al., 2013)	18	400	140	0,1
Abalone	10	3133	1044	0,1,2,3
Cardiotocography	30	1126	1000	0,1,2
WineRed	11	1100	499	0,1,2
WineWhite	11	3000	1898	0,1,2

Table 2.1: Dataset description showing the number of covariates, the number of training points, the number of testing points and the levels of responses for each dataset.

To decide the generative distribution of covariates before running our algorithm, we perturb the empirical distribution by Gaussian noise whose variances are approximately 1/50 of the ranges of corresponding covariates. Probability of jumping to neighboring category for discrete covariates is set to be 1/7.

We compare across four methods here: classification trees (CART), random forests (RF), our proposed approximating tree (AppTree), and a baseline method (BASE). Previous work (Johansson and Niklasson, 2009; Johansson et al., 2010) fixes the number of

pseudo sample points from the oracle during the coaching procedure. Analogously, we set BASE to be a non-adaptive version of our AppTree which requests a pseudo sample set from the RF only once at the root node and uses it all the way down. The pseudo sample size is set to be 9 times the size of the training data, which is larger than what was being used in (Johansson and Niklasson, 2009; Johansson et al., 2010) and is designed as a reasonable blind decision without any prior information.

We use the same setting for all datasets for consistency. For each dataset, we train a RF containing 200 trees, a CART, then 100 AppTrees and 100 BASEs iteratively approximating the RF. $N_{ps} = 500,000$, which means each node of AppTree can generate approximately at most 5×10^5 pseudo sample points to decide its split. CART, BASE and AppTree all grow to the 6th layer including the root. Confidence level α is set to be 0.1.

2.4.3 Binary Classification

Figure 2.4 shows the evaluation of methods on binary classification datasets. Johansson et al. (2011) has pointed out that a single decision tree is already capable of mimicking an oracle predictor (the teacher) to make highly accurate predictions given the oracle is not overly complicated. Our simulation shows similar results, as all three methods CART, BASE and AppTree tightly follow the ROC curves of the RF and there is no significance difference among them. Consistency is measured as the frequency of a model agreeing with the RF when predicting on same input covariates. We use a moving threshold as the classification bound, and evaluate the consistency on both the testing data and the

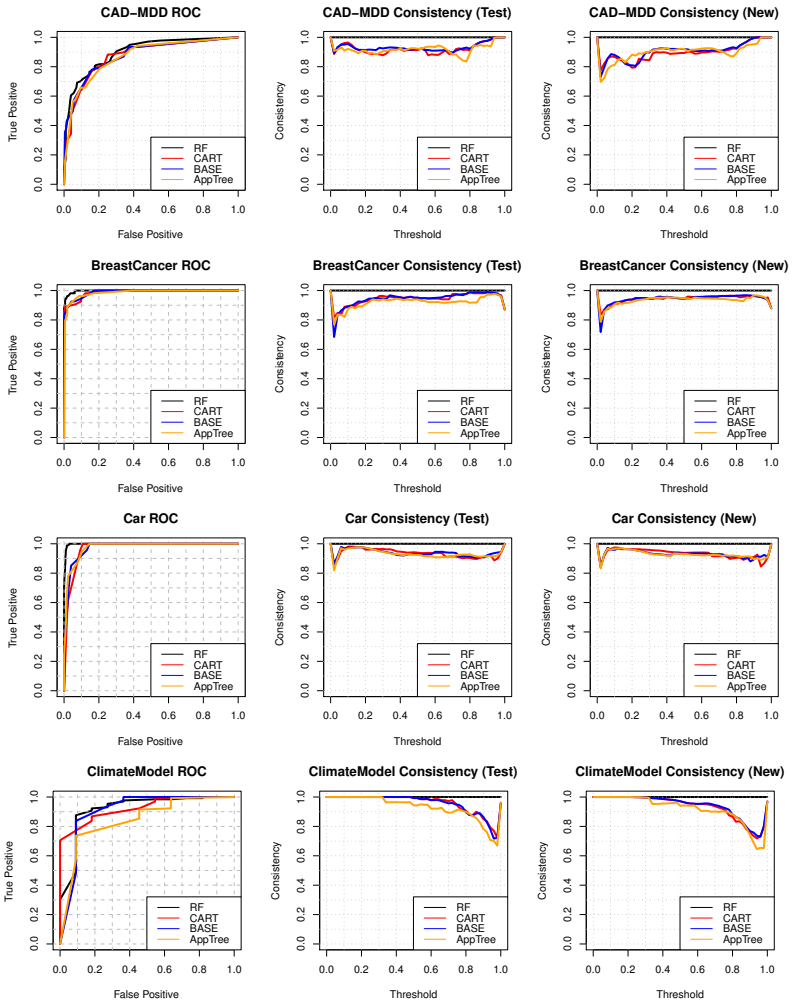


Figure 2.4: Performance evaluation on binary classification datasets. From top to bottom: CAD-MDD, BreastCancer, Car, ClimateModel. From left to right: ROC curves, consistency with RF on testing set, consistency with RF on new data points.

extrapolated new data (as marked "test" and "new" in the plot). First, all three 6-layer trees can approximate the RF with over 80% probability for almost any given classification threshold, and there is no significant difference among them. Further, the behaviors of both "test" and "new" plot seem similar, which shows support to our generative covariate distribution estimation. While the overall 80% consistency may seem not powerful enough to make those trees aligned with the oracle RF, we can build the trees deeper to better "overfit" the RF.

2.4.4 Multiclass Classification

Figure 2.5 demonstrates the evaluation on 3 multiclass classification datasets. We observe similar patterns as we did in binary cases that all three tree methods work similarly. ROC curves of RF are less ideal this time, and ROC curves of three tree methods are a bit off, which is a sign that deeper trees might be necessary. In general, it is reasonable to believe that by extending the tree we could approximate a given black box as accurate as possible, especially when the black box is a RF which shares with trees the similar pattern to orthogonally segment the covariate space. On the other hand, we also expect different black box prediction functions when any shallow tree approximation should not be effective to approximate the RF.

In terms of consistency, all tree methods are again capable of agreeing with the RF on about 80% of the predictions made by RF on the testing data. We have therefore shown that our stability request of AppTree does not undermine its predictive power and consistency

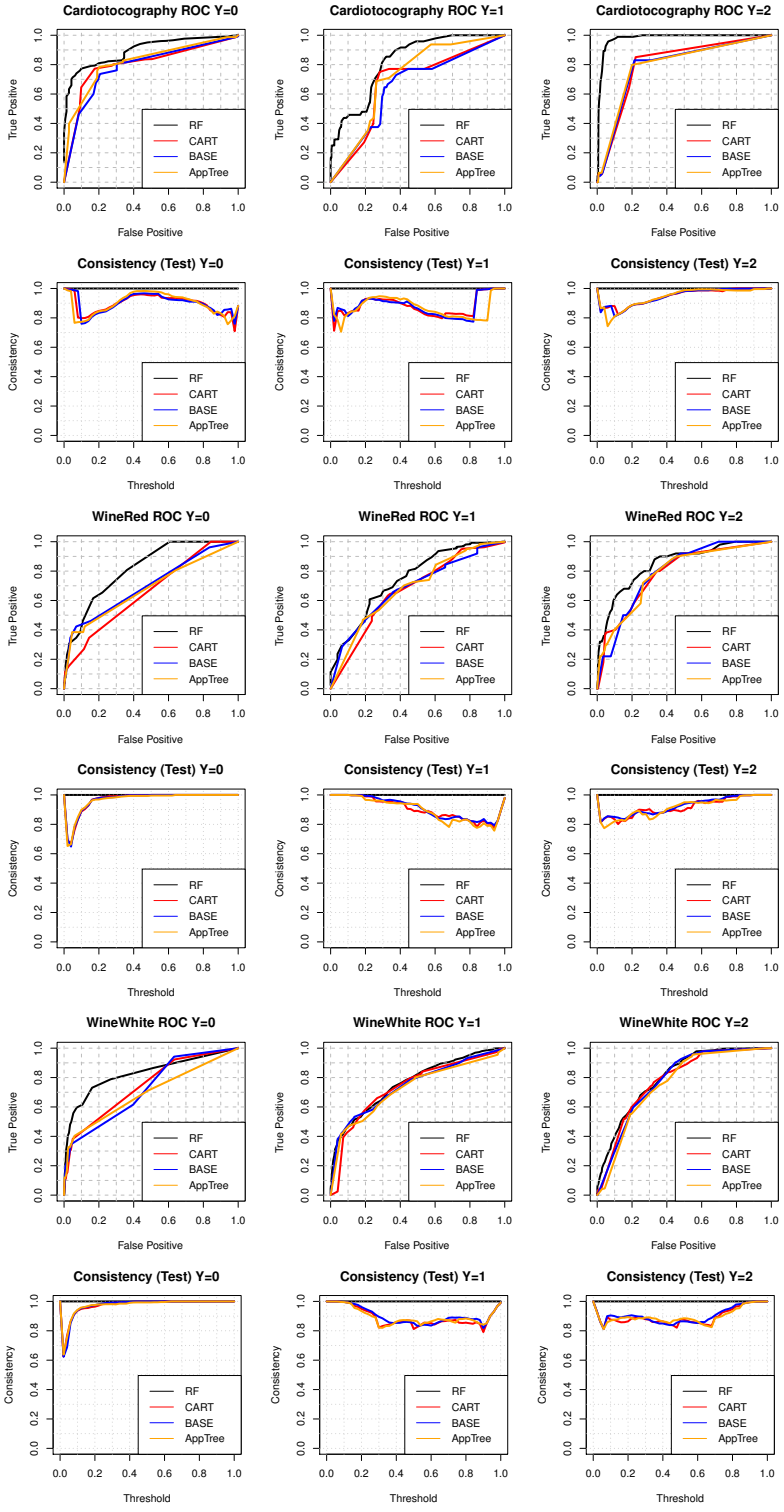


Figure 2.5: Performance evaluation on multiclass (3-class) classification datasets. ROC curves are plotted in a one v.s. all fashion. Consistency is only checked on testing data. From top to bottom: Cardiotocography, WineRed, WineWhite.

with the black box when compared with other tree methods.

2.4.5 Stability

Stability is the major concern in our simulation study. We measure how many different tree structures BASE and AppTree report out of their 100 replications of approximating the same RF, and count how many times each individual tree structure (both splitting covariate and splitting value) occurs. Table 2.2 shows the number of different tree structures and number of occurrences of the top 3 frequent structures for both BASE and AppTree on each dataset. Figure 2.6 and 2.7 visualize the results.

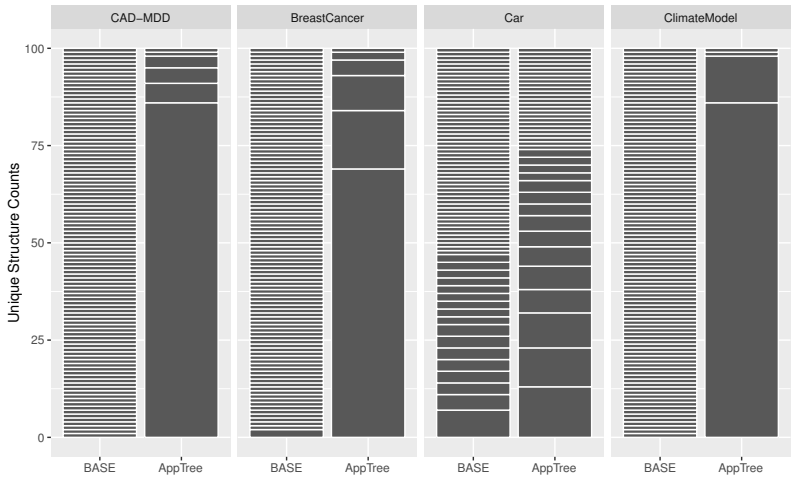


Figure 2.6: BASE and AppTree stability measured on binary classification datasets. From left to right: CAD-MDD, BreastCancer, Car, ClimateModel. In each column, a single black bar represents a unique structure of the tree, while the height of the bar represents the number of occurrence of that structure out of 100 replications.

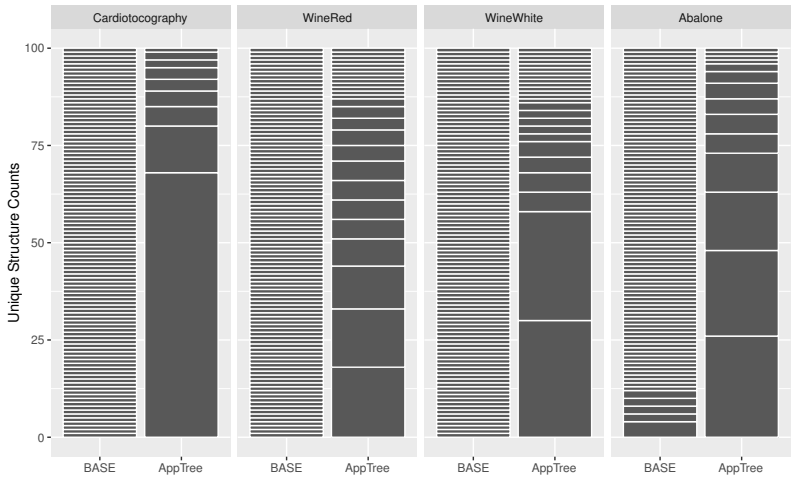


Figure 2.7: BASE and AppTree stability measured on multiclass classification datasets. From left to right: Cardiotocography, WineRed, WineWhite, Abalone. In each column, a single black bar represents a unique structure of the tree, while the height of the bar represents the number of occurrence of that structure out of 100 replications.

BASE is supposed to be a non-adaptive version of AppTree that only requests pseudo sample points once at the root node. Our simulation setting guarantees that BASE and AppTree have access to the same set of all possible splitting covariates and values. If we compared AppTree with BASE equipped with an enormous amount of pseudo sample points at the beginning such that at each node BASE had no fewer sample points than AppTree, we should expect similar behavior between those two methods. However, BASE fails to stabilize the tree structure in our experiment as almost every 6-layer tree it produces has an identical structure, whereas AppTree manages to generate a small number of dominant tree structures with a confidence control of $\alpha = 0.1$. It proves that our test of better split and adaptive increment of pseudo sample size significantly contribute to the stability of the decision tree we obtain from the coaching procedure as the approximating tree.

Name	BASE		AppTree	
	#Struct	Top 3 Cnt	#Struct	Top 3 Cnt
CAD-MDD	100	1, 1, 1	6	**86**, 5, 4
BreastCancer	99	2, 1, 1	6	**69**, 15, 9
Car	70	7, 4, 3	41	**13**, 10, 9
ClimateModel	100	1, 1, 1	4	**86**, 12, 1
Abalone	93	4, 2, 2	14	**26**, 22, 15
Cardiotocography	100	1, 1, 1	9	**68**, 12, 5
WineRed	100	1, 1, 1	26	**18**, 15, 11
WineWhite	100	1, 1, 1	25	**30**, 28, 5

Table 2.2: Stability of BASE and AppTree. The table shows the number of identical structures out of 100 replications and counts the occurrences of the top 3 structures in each case. Cnt for counts. Boldfaced numbers show the occurrences of the dominant tree structure out of 100 replications generated by AppTree for each dataset.

Notice that $0.95^{31} \approx 0.2$, which means if we choose $\alpha = 0.05$ and train with with infinitely many pseudo sample points, we should have the most dominant 6-layer tree structure occurring about 20 out of 100 replications. Our results on most of the datasets have already attained such stability with $\alpha = 0.1$ and $N_{ps} = 5 \times 10^5$, therefore the control of α is relative conservative while the choice of the pseudo sample cap $N_{ps} = 5 \times 10^5$ is sufficient. The significance level α controls the stability at a split-wise level. It is possible to extend this to further stabilize the tree by again introducing the FWER at the tree level. Notice this procedure may also increase the number of pseudo sample points we need at each split.

2.5 Model Fitting v.s. Distillation

Our theory simulation study suggests the need for a massive number of pseudo sample points to optimize a single split in the context of tree distillation. However, such scenario also applies to the ordinary CART training when we collect data and fit decision trees to classify or regress, while the teacher model now is the actual underlying distribution on the covariate space and the means to inquire the teacher model is through experimental design and data collection. Our results provide the evidence that decision tree splits are unstable without the presence of big training sets. This relates to the trade off between sample size, model stability and model interpretability. We will show in later chapters that the standard CART building strategy with Gini indices is quite flawed in terms of allowing mathematical analyses because of its inherent greediness which repels an analyzable mathematical description. We have to choose between two factions: either to follow standard tree method and utilize big samples, or to embrace uncertainties in trees and gain mathematical advantages.

However, certain aspects of constructing a stable tree distillation are not particularly tethered to decision trees, whereas they can be treated as alternative difficulty measures or model selection criteria for mode fitting. For example, when a model distillation can at the same time achieve the predictive accuracy and the coherence of a given black box, it implies that either the black model works as an interpretable glass box, or the underlying learning task is too simple for the chosen black box. Notice the crucial difference between contrasting models trained by a black box and a glass box simultaneously, and analyzing a glass box model trained stably as the distillation from a black box model. The latter pays

more attention to evaluating the inherent behavior of the black box, therefore can be used to evaluate how much a chosen black box matches the learning problem.

CHAPTER 3

BOULEVARD BOOSTED TREES AND THEIR ASYMPTOTICS

3.1 Gradient Boosted Decision Trees and Boulevard

Analyses of RFs have relied on a subsampling structure to express the estimator in the form of a U-statistic from which central limit theorems can be derived. By contrast, GBDT produces trees sequentially with the current tree depending on the values in those built previously, requiring a different analytical approach. While the algorithm proposed in Friedman (2001) is intended to be generally applicable to any loss function, in this chapter we focus specifically on nonparametric regression (Stone, 1977, 1982). Given a sample of n observations $(x_1, y_1), \ldots, (x_n, y_n) \in [0,1]^d \times \mathbb{R}$, assume they follow the relation

$$X \sim \mu, \quad Y = f(X) + \epsilon$$

which satisfies the following:

(M1) μ the density is bounded from above and below, i.e. $\exists 0 < c_1 < c_2$ s.t. $c_1 \leq \mu \leq c_2$.

(M2) f is bounded Lipschitsz, i.e. $|f(x)| \leq M_f < \infty$, and $\exists \alpha > 0$ s.t. $|f(x_1) - f(x_2)| \leq \alpha |x_1 - x_2|, \forall x_1, x_2 \in [0,1]^d$.

(M3) ϵ is sub-Gaussian error with $\mathbb{E}[\epsilon] = 0$, $\mathbb{E}\left[\epsilon^2\right] = \sigma_\epsilon^2$, $\mathbb{E}\left[\epsilon^4\right] < \infty$.

GBDT builds correlated trees in a sequential fashion so that each tree predicts the gradient of current training error so as to perform gradient descent in functional space

(Friedman et al., 2000). A typical GBDT estimating $\hat{f} = \mathbb{E}[Y|X]$, is represented as a tree ensemble version of the Robbins-Monro algorithm (Robbins and Monro, 1951), and combines standard GBDT with L^2 loss leading to an iterative fitting of residuals. The procedure is given as

Algorithm 3.1 (GBDT)**.**

- *start with* $\hat{f}_0 = 0$*;*
- *For* $b = 1, \ldots,$ *given* $\hat{f}_b$*, calculate the gradient*

$$z_i \triangleq -\frac{\partial}{\partial u_i} \sum_{i=1}^{n} \frac{1}{2} (u_i - y_i)^2 \Big|_{u_i = \hat{f}_b(x_i)} = y_i - \hat{f}_b(x_i);$$

- *construct a tree regressor* $t_b(\cdot)$ *on* $(x_1, z_1), \ldots, (x_n, z_n)$*;*
- *update by a small learning rate* $\lambda > 0$*,*

$$\hat{f}_{b+1} = \hat{f}_b + \lambda t_b.$$

Gradient boosting developed from attempts to understand adaboost (Freund et al., 1999) in Friedman et al. (2000). Mallat and Zhang (1993) studied the Robbins-Monro algorithm and showed the convergence when the additive components are taken from a Hilbert space. As for the tree version of the Robbins-Monro algorithm, Bühlmann (2002) showed the consistency under L^2 norm. From a broad point of view, discussions on consistency and convergence of general L^2 boosting framework can be found in Bühlmann and Yu (2003), Zhang et al. (2005) and Bühlmann and Hothorn (2007).

There are a number of variations on the algorithm presented above. Friedman (2002) incorporated subsampling in each iteration and empirically showed significant improvement in predictive accuracy. Rashmi and Gilad-Bachrach (2015) argued that GBDT is sensitive towards the beginning, requiring lots of later trees to make an impact. They borrowed the idea of dropout (Wager et al., 2013; Srivastava et al., 2014) which trains and weighs each new iteration with a subset of the existing ensemble to handle such imbalance which they called "over specification". Similarly, Rogozhnikov and Likhomanenko (2017) suggested to sequentially scale down the learning rate and studied the convergence of the boosting path when the learning rate is small enough to guarantee contraction.

All methods mentioned above attempt to regularize boosting to avoid excessive dependence on the initial trees in the ensemble which may lead GBDT to be trapped in local minima. We hope to unify those methods by carefully combining both subsampling and adaptive learning rate shrinkage into gradient boosted trees to study its asymptotic behavior, leading to a predictive model capable of statistical inference.

This chapter is particularly inspired by the recent development of the RF inferential framework (Mentch and Hooker, 2016; Wager and Athey, 2017; Mentch and Hooker, 2017), in which the averaging structure of random forests results in an analysis based on U-statistics and Hájek projection leading to the asymptotic normality. Similarly, in classic stochastic gradient methods, Ruppert-Polyak (Polyak and Juditsky, 1992; Ruppert, 1988) averaging is used in achieving asymptotic normality for model parameter estimators by averaging the gradient descent history. The boosting framework we present results in a model that also exhibits this averaging structure which we can therefore leverage.

To contrast the sequential development in GBDT with RF we have named this algorithm *Boulevard*.

Because of the mathematical difficulties of analyzing the greedy splitting rules of trees, most current analyses of RFs have been based on variations of the procedure originally proposed in Breiman (2001). Both Mentch and Hooker (2016) and Wager and Athey (2017) replace bootstrap sampling with subsampling. Wager and Athey (2017) also imposes an honesty condition via subsample splitting to make the tree structure independent of leaf values. While these may improve performance, other simplifications such as the use of completely randomized trees are unlikely to be practically useful, but did allow the development of initial consistency results in Biau (2012) and a connection to kernel methods in Davies and Ghahramani (2014) and Scornet (2016). In a similar fashion, we believe that the use of subsampling and shrinkage are important for our results. However, we also assume a global independence between tree structures and leaf values which we term "non-adaptivity". We think this condition can be relaxed and that doing so is important for the performance of Boulevard.

So far as we are aware, these represent the first results on a distributional limit for GBDT and hence the potential for inference using this framework; we hope that they inspire further refinements. Bayesian Additive Regression Trees (BART) (Chipman et al., 2010) were also motivated by GBDT and allow the development of Bayesian credible intervals. However, the training procedure for BART resembles backfitting a finite number of trees, resulting in a somewhat different model class. Nonetheless, we expect that some of the stochastic contraction mapping results developed below may be useful in demon-

strating frequentist properties for the resulting BART estimators.

3.1.1 Boulevard

Algorithm 3.2 provides a formal statement of the Boulevard algorithm. This incorporates both subsampling and on-the-fly shrinkage into GBDT.

Algorithm 3.2 (Boulevard)**.**

- *Start with* $\hat{f}_0 = 0$*.*
- *Given* $\hat{f}_b$*, calculate the gradient*

$$z_i \triangleq -\frac{\partial}{\partial u_i} \sum_{i=1}^{n} \frac{1}{2} (u_i - y_i)^2 \Big|_{u_i = \hat{f}_b(x_i)} = y_i - \hat{f}_b(x_i). \tag{3.1}$$

- *Generate a subsample* $w \subset \{1, 2, \ldots, n\}$*.*
- *Construct a tree regressor* $t_b(\cdot)$ *on* $\{(x_i, z_i), i \in w\}$*.*
- *Update by learning rate* $1 > \lambda > 0$*,*

$$\hat{f}_{b+1} = \frac{b-1}{b} \hat{f}_b + \frac{\lambda}{b} t_b = \frac{\lambda}{b} \sum_{i=1}^{b} t_i.$$

This design transforms the ensemble to be an average over all trees instead of continually adding trees together. The benefit of this is twofold. First, shrinkage makes the ensemble less sensitive to any particular tree. It leaves part of the signal in the gradient guaranteeing that no tree is fit to entire error. Second, subsampling reduces overfitting. As

a result, the final form of the predictor sits between an ordinary GBDT and a random forest. The name Boulevard comes from the fact that during construction, older trees shrink but all trees are eventually of equal importance, just as if we were walking on a boulevard and looking backwards.

3.2 Honest Trees and Forests

3.2.1 Honest Trees and Honest Forests

We illustrate in this section the construction of base tree leaners in the Boulevard algorithm. A decision tree (Breiman et al., 1984) predicts by iteratively segmenting the covariate space into disjoint subsets (i.e. leaves) within each of which the average (or the majority vote) of observations serves as the leaf value. Therefore we can represent a regression tree as a linear combination of observations.

Suppose a regression tree $t_n(\cdot)$ segments certain covariate space Ω into a disjoint union $\Omega = \bigsqcup_{j=1}^{m} A_j$. We also refer to $\{A_j\}_{j=1}^{m}$ as the leaves or the tree structure. In our case, $\Omega = [0,1]^d$ and $\{A_j\}_{j=1}^{m}$ hyper-rectangles. We explicitly express $t_n(\cdot)$ as

$$t_n(x) = \sum_{i=1}^{n} s_{n,i}(x) y_i,$$

where, given $x \in A_j$,

$$s_{n,k}(x) = \frac{I(x_k \in A_j)}{\sum_{i=1}^{n} I(x_i \in A_j)}.$$

Slight changes should be made to this expression when a subsample is used instead of the full sample to calculate the leaf value. For given subsample $w \subset \{1, \ldots, n\}$, we write

$$t_n(x; w) = \sum_{i=1}^{n} s_{n,i}(x; w) y_i.$$

In this case, for any $x \in A_j$,

$$s_{n,k}(x) = s_{n,k}(x; w) = \frac{I(x_k \in A_j)}{\sum_{i=1}^{n} I(x_i \in A_j) I(i \in w)} = \frac{I(x_k \in A_j) I(k \in w)}{\sum_{x_i \in A_j} I(i \in w)}.$$

In both cases, we call $s_n(x) = (s_{n,1}(x), \ldots, s_{n,n}(x))^T$ the (column) *structure vector* of x, and

$$S_n = \begin{bmatrix} s_{n,1}(x_1) & \ldots & s_{n,n}(x_1) \\ \vdots & \ddots & \\ s_{n,1}(x_n) & \ldots & s_{n,n}(x_n) \end{bmatrix} = \begin{bmatrix} s_n(x_1)^T \\ \vdots \\ s_n(x_n)^T \end{bmatrix}$$

the *structure matrix* as the stacked structure vectors of the sample.

The greedy algorithms typically used to build decision trees have proved particularly challenging for mathematical analysis. It is difficult to provide guarantees that it will not isolate sample points with large observation errors, i.e. outliers, thereby de-stabilizing the resulting predicted values. We describe this behavior as "chasing order statistics". As a result, most results on trees and tree ensembles rely on randomization, for example, using completely randomized splits or retaining a small chance of making randomized split covariates (Bühlmann et al., 2002; Biau, 2012; Scornet, 2016; Wager and Athey, 2017).

In particular, Wager and Athey (2017) introduced the concept of *honesty* through double-sample trees which apply two different subsamples: one to decide tree structure and another to calculate leaf values. While this strategy allows the sample to determine

the tree structure, it creates conditional independence between the tree structure and the leaf values to prevent trees from being doubly influenced by clustered outliers. In a similar manner, our analysis requires stringent isolation between these two steps. One way to achieve so is by not looking at the training responses while deciding the tree structure, as shown in the second step of the clarification of our honest tree strategy with subsampling given in Algorithm 3.3.

Algorithm 3.3 (Honest Trees)**.**

- *Start with a sample of size n,* $(x_1, y_1), \ldots, (x_n, y_n)$.
- *Obtain the tree structure* $q = \{A_j\}_{j=1}^{m}$ *independently of* $y_1, \ldots, y_n$.
- *Uniformly subsample an index set* $w \in \{1, \ldots, n\}$ *of size* θn.
- *Decide the leaf values, hence* $t_n(\cdot)$*, merely w.r.t w as for* $x \in A_j$*,*

$$t_n(x) = \sum_{x_i \in A_j} \frac{I(i \in w)}{\sum_{x_l \in A_j} I(l \in w)} \cdot y_i,$$

  with $0/0$ *defined to be* 0.

However, a disadvantage of honest trees is the possibility that there could be no subsample points in a terminal leaf when deciding the leaf values by the second subsample. We choose to predict 0 for expediency, in which case the corresponding tree structure vector for points in such leaf will be zeroes. We refer to this issue as *missing terminal subsample* and will later show that it can be avoided asymptotically by selecting a sufficiently large subsample rate.

The following theorem shows the properties we obtain by applying the honest tree strategy. One major contribution of honesty is the symmetry of the expected structure matrix, which connects it to the kernel form of a subsample decision tree.

Theorem 3.1. *Denote $\mathbb{E}_w$ as the expectation over all possible subsample index sets. For a fixed segmentation (tree structure) $q = \{A_j\}_{j=1}^m$,*

(i) $\mathbb{E}_w[S_n]$ *is element-wisely nonnegative, symmetric.*

(ii) $\mathbb{E}_w[S_n]$ *is positive semi-definite.*

(iii) $\|\mathbb{E}_w[S_n]\| \leq 1$.

We now move from a single tree to a tree ensemble, starting from random forests (Breiman, 2001). The concept of subsampling and bagging has been intensely used in the construction of random forests whose component trees have distinct structures due to the random set of sample points and splitting covariates. Denote by $(\mathcal{Q}_n, \mathcal{Q}_n)$ the probability space of all possible tree structures given sample $(x_1, y_1), \ldots, (x_n, y_n)$ of size n and an approach of deciding tree structures with randomness, where $q = \{A_i\}_{i=1}^{m_q} \in \mathcal{Q}_n$ is the structure of a single possible tree. On one hand, if each tree in the forest is honest, we could write the expected random forest prediction on the sample as

$$\hat{Y} = \mathbb{E}_q[\mathbb{E}_w[S_n]] \cdot Y = \mathbb{E}_{q,w}[S_n] \cdot Y,$$

where $Y = (y_1, \ldots, y_n)^T$ and $\mathbb{E}_q$ the expectation w.r.t. probability measure $\mathcal{Q}_n$. On the other hand, supposing we build a single honest tree deciding tree structure from the structural space $\mathcal{Q}_n$ with probability measure $\mathcal{Q}_n$, $\mathbb{E}_{q,w}[S_n]$ is also the expected structure matrix which carries most properties of $\mathbb{E}_w[S_n]$.

Corollary 3.2. *Denote $\mathbb{E}_{q,w}$ as the expectation over all possible tree structures and subsample index sets, then*

(i) $\mathbb{E}_{q,w}[S_n]$ *is symmetric, element-wisely nonnegative.*

(ii) $\mathbb{E}_{q,w}[S_n]$ *is positive semi-definite.*

(iii) $\left\|\mathbb{E}_{q,w}[S_n]\right\| \leq 1$.

Here $\mathbb{E}_{q,w}[S_n]$ is similar to the random forest kernel defined by the corresponding tree structure space, subsampling strategy and tree structure randomization approach.

3.2.2 Adaptivity of Boosted Trees

As mentioned above, when building a random forest, the current ensemble has no influence on either the structure or the leaf values of the following trees. We could also imagine an ideal boosting scenario that has reached stationarity, after which all subsequent trees should behave identically regardless of the current ensemble. One common property is that the distribution of tree structures should be identical across trees. We refer to this property as the *(non)-adaptivity* of tree ensembles, which is defined formally as follows.

Definition 3.1. Denote $(Q_{n,b}, \mathcal{Q}_{n,b})$ the probability space of all possible tree structures given sample $(x_1, y_1), \ldots, (x_n, y_n)$ of size n after b trees have been built. A tree ensemble is *non-adaptive* if $(Q_{n,b}, \mathcal{Q}_{n,b})$ is identical across b. A tree ensemble is *eventually non-adaptive* if $(Q_{n,b}, \mathcal{Q}_{n,b})$ is identical for sufficiently large b.

The non-adaptivity of random forests contributes to the convenience of taking expectation of the ensemble since all trees are independent and identically distributed. In contrast, conventional gradient boosted trees are adaptive. For each new tree, both the structure and the leaf values use the latest gradient that changes along with the growing ensemble. As a result, any analysis has to condition on the current ensemble state. Honesty and non-adaptivity resolve this issue on different levels. In terms of a single decision tree, building an honest tree helps to reduce the dependence by untying the tree structure from the gradient. In terms of the entire tree ensemble, non-adaptivity further simplifies the analysis that we use a shared tree structure space and distribution.

In contrast, eventual non-adaptivity is a necessary condition should boosting predictions become stationary after enough iterations. We will discuss the details in Section 5.

In practice, there are a few possible means to enforce non-adaptivity by deciding all tree structures independently of the gradient. One is through completely randomized trees for which the gradient only influences the leaf values. An alternative strategy is to acquire another independent sample $(x'_1, y'_1), \ldots, (x'_n, y'_n)$ solely for determining tree structures. We will refer to the Boulevard algorithm equipped with this mechanism as *non-adaptive Boulevard* for the rest of the thesis.

3.3 Boulevard Convergence

Following from Zhang et al. (2005), a first heoretical issue of analyzing boosting method is the difficulty of attaining convergence. As a starting point we will show that Boulevard guarantees point-wise convergence under finite sample settings.

3.3.1 Stochastic Contraction and Boulevard Convergence

To prove convergence of the Boulevard algorithm, we introduce the following definition, lemmas and theorem inspired by the unpublished manuscript by Almudevar (Almudevar) regarding a special class of stochastic processes. We refer the readers to the original manuscript, but key points of the proof are briefly reproduced and extended here for the study of Boulevard asymptotics.

Theorem 3.3 (Multidimensional Stochastic Contraction)**.** *Given* $\mathbb{R}^d$ *stochastic process* $\{Z_t\}_{t\in\mathbb{N}}$*, a sequence of* $0 < \lambda_t \leq 1$*, define*

$$\mathcal{F}_0 = \emptyset, \mathcal{F}_t = \sigma(Z_1, \ldots, Z_t),$$

$$\epsilon_t = Z_t - \mathbb{E}\left[Z_t | \mathcal{F}_{t-1}\right].$$

We call Z_t *a (multidimensional) stochastic contraction if the following properties hold*

(C1) Vanishing coefficients

$$\sum_{t=1}^{\infty}(1 - \lambda_t) = \infty, \text{ i.e. } \prod_{t=1}^{\infty}\lambda_t = 0.$$

(C2) Mean contraction

$$\|\mathbb{E}\left[Z_t|\mathcal{F}_{t-1}\right]\| \leq \lambda_t \|Z_{t-1}\|, a.s..$$

(C3) Bounded deviation

$$\sup \|\epsilon_t\| \to 0, \quad \sum_{t=1}^{\infty} \mathbb{E}\left[\|\epsilon_t\|^2\right] \leq \infty.$$

In particular, a multidimensional stochastic contraction exhibits the following behavior

(i) Contraction

$$Z_t \xrightarrow{a.s.} 0.$$

(ii) Kolmogorov inequality

$$P\left(\sup_{t\geq T} \|Z_t\| \leq \|Z_T\| + \delta\right) \geq 1 - \frac{4\sqrt{d}\sum_{t=T+1}^{\infty}\mathbb{E}\left[\epsilon_t^2\right]}{\min\{\delta^2, \beta^2\}}, \tag{3.2}$$

where $\beta = \|Z_T\| + \delta - \sqrt{d}\sup_{t>T}\|\epsilon_t\| > 0$.

The proof is provided in Appendix 3.7.2. The Kolmogorov inequality, which is novel from the original manuscript, is a direct corollary from the original proof in Almudevar (Almudevar).

Working with non-adaptive Boulevard, adaptive shrinkage grants it the structure of a stochastic contraction. We now apply Theorem 3.3 to show the convergence.

Theorem 3.4. *Given sample* $(x_1, y_1), \ldots, (x_n, y_n)$. *If we construct gradient boosted trees non-adaptively with identical tree structure space* $(Q_n, \mathcal{Q}_n)$ *and honest regression trees, by*

choosing $M \gg \max\{M_f, y_1, \ldots, y_n\}$ *and defining* $\Gamma_M(x) = sign(x)(|x| \wedge M)$ *as a truncation function, let Boulevard iteration take form of*

$$\hat{f}_b(x) = \frac{b-1}{b}\hat{f}_{b-1}(x) + \frac{\lambda}{b}s_b(x)(Y - \Gamma_M(\hat{Y}_{b-1})), \tag{3.3}$$

where $Y = (y_1, \ldots, y_n)^T$ *the observed response vector,* $\hat{Y}_b = (\hat{f}_b(x_1), \ldots, \hat{f}_b(x_n))^T$ *the predicted response vector by the first b trees,* s_b *the random tree structure vector. Hence*

$$\hat{Y}_b \longrightarrow \left[\frac{1}{\lambda}I + \mathbb{E}\left[S_n\right]\right]^{-1} \mathbb{E}\left[S_n\right] Y,$$

where $\mathbb{E}\left[\cdot\right] = \mathbb{E}_{q,w}\left[\cdot\right]$, S_n *the random tree structure matrix defined above.*

Proof. Due to non-adaptivity S_n is independent of $\hat{Y}_b$ for any b. Notice that $Y^* = \lambda\mathbb{E}\left[S_n\right](Y - Y^*)$ for $Y^* = \left[\frac{1}{\lambda}I + \mathbb{E}\left[S_n\right]\right]^{-1}\mathbb{E}\left[S_n\right]Y$. Define the filtration $\mathcal{F}_b = \sigma(\hat{Y}_0, \ldots, \hat{Y}_b)$ and consider the sequence $Z_b = \hat{Y}_b - Y^*$. This sequence satisfies the stochastic contraction condition. First, $\|Z_0\| = \|Y^*\| \leq \infty$. Notice

$$\begin{aligned}
\|\mathbb{E}\left[Z_b|\mathcal{F}_{b-1}\right]\| &= \left\|\mathbb{E}\left[\frac{b-1}{b}\hat{Y}_{b-1} + \frac{\lambda}{b}S_n(Y - \Gamma_M(\hat{Y}_{b-1})) - Y^*\middle|\mathcal{F}_{b-1}\right]\right\| \\
&= \left\|\frac{b-1}{b}(\hat{Y}_{b-1} - Y^*) + \frac{\lambda}{b}\mathbb{E}\left[S_n\right](Y - \Gamma_M(\hat{Y}_{b-1})) - \frac{1}{b}Y^*\right\| \\
&\leq \frac{b-1}{b}\left\|\hat{Y}_{b-1} - Y^*\right\| + \left\|\frac{\lambda}{b}\mathbb{E}\left[S_n\right](Y - \Gamma_M(\hat{Y}_{b-1})) - \frac{\lambda}{b}\mathbb{E}\left[S_n\right](Y - Y^*)\right\| \\
&\leq \frac{b-1+\lambda}{b}\left\|\hat{Y}_{b-1} - Y^*\right\| \triangleq k_b\left\|Z_{b-1}\right\|,
\end{aligned}$$

where $\sum_{b=1}^{\infty}(1 - k_b) = \infty$. Since entries and row sums of are both ≤ 1,

$$\|S_n\| \leq \sqrt{\|S_n\|_\infty \|S_n\|_1} \leq \sqrt{1 \times n} = \sqrt{n}.$$

Therefore

$$\|\epsilon_b\| = \|Z_b - \mathbb{E}\left[Z_b|\mathcal{F}_{b-1}\right]\| = \left\|\frac{\lambda}{b}(\mathbb{E}\left[S_n\right] - S_n)(Y - \Gamma_M(\hat{Y}_{b-1}))\right\| \leq \frac{\lambda}{b}(1 + \sqrt{n})2\sqrt{n}M.$$

Hence

$$\sum_{b=1}^{\infty} \mathbb{E}\left[\|\epsilon_b\|^2\right] \leq \left(\sum_{b=1}^{\infty} \frac{1}{b^2}\right) \cdot \lambda^2(1 + \sqrt{n})^2 4nM < \infty.$$

We conclude that $Z_b \xrightarrow{a.s.} 0$, i.e. $\hat{Y}_b \xrightarrow{a.s.} Y^*$. □

This theorem guarantees the convergence of Boulevard path under finite sample setting once we threshold it by a large M. Non-adaptivity serves here to decompose every tree model into the multiplication of an independent structure matrix and a predictable response vector.

As a corollary we obtain the expression of the prediction at any point of interest x. The result takes the form of a kernel ridge regression with the random forest kernel (Scornet, 2016).

Corollary 3.5. *By defining* $\hat{f} = \lim_{b\to\infty} \hat{f}_b$,

$$\hat{f}(x) = \mathbb{E}\left[s_n(x)\right] \left[\frac{1}{\lambda} I + \mathbb{E}\left[S_n\right]\right]^{-1} Y. \tag{3.4}$$

Ridge regression tends to shrink the predictions towards 0 and so does (3.4). The iterative averaging of Boulevard algorithm along with λ results in Boulevard predictions covering $\frac{\lambda}{1+\lambda}$ of the signal instead of the full signal. We will prove and discuss in details this behavior in Section 3.4.5.

3.3.2 Beyond L^2 Loss

Besides regression, other tasks may require alternative *loss* functions for boosting, for instance, the exponential loss $L(w, y) = \exp(-wy)$ in adaboost (Freund and Schapire, 1995). Analogous to the proof for L^2 loss, we can write the counterparts for any general loss $L(u) = \sum_i L(u_i, y_i)$ whose non-adaptive Boulevard iteration takes the form of

$$\hat{Y}_b = \frac{b-1}{b}\hat{Y}_{b-1} - \frac{\lambda}{b} S_n \nabla_w L(w)\Big|_{w=\hat{Y}_{b-1}}.$$

Suppose the existence of the fix point $\hat{Y}^* = -\lambda \mathbb{E}[S_n] \nabla_w L(w)\Big|_{w=\hat{Y}^*}$, then

$$\mathbb{E}\left[\hat{Y}_b - \hat{Y}^* | \mathcal{F}_{b-1}\right] = \frac{b-1}{b}(\hat{Y}_{b-1} - \hat{Y}^*) - \frac{\lambda}{b}\mathbb{E}[S_n]\left(\nabla_w L(w)\Big|_{w=\hat{Y}_{b-1}} - \nabla_w L(w)\Big|_{w=\hat{Y}^*}\right).$$

If the gradient term is bounded and Lipschitz (which could be enforced by truncation), i.e.

$$\left\| \nabla_w L(w)\Big|_{w=w_1} - \nabla_w L(w)\Big|_{w=w_2} \right\| \leq M \|w_1 - w_2\|,$$

we can similarly show such Boulevard iteration converges by choosing $\lambda \leq M^{-1}$. However, the closed form of $\hat{Y}^*$ can be intractable to obtain and potentially non-unique. For example for AdaBoost, $\hat{Y}^*$ is the solution to $\hat{Y}^* = -\lambda \mathbb{E}[[]\, S_n](\exp(-\hat{Y}_1^* y_1), \ldots, \exp(-\hat{Y}_n^* y_n))^T$.

3.4 Asymptotic Normality

Inspired by recent results demonstrating the asymptotic normality of random forest predictions, in this section we prove the asymptotic normality of predictions from Boulevard. Before detailing these results, we need some prerequisite discussion on the rates used for

decision tree construction in order to ensure asymptotic local behavior. In general, the variability of model predictions comes from two sources: the variability of the random sample we use to train the model, and the variability of the response errors. The strategy for our proof is as follows: we first consider the *fixed design* case where the sequence of increasing samples are supposedly determined and have the properties we require, so only the response errors contribute to the variability. We then establish the uniformity over almost all random sample sequences to extend the limiting distribution to *random design* cases, showing that it is still the response errors that dominate the prediction variability.

3.4.1 Building Deeper Trees

Decision trees can be thought as k-nearest-neighbor (k-NN: Altman, 1992) models where k is the leaf size and the distance metric is given by whether two points are in the same leaf. This adapts the metric to the local geometry of the response function. As the conclusions on k-NN predictions require growing-in-size and shrinking-in-radius neighborhoods (Gordon and Olshen, 1984), so are the counterparts of building deeper trees. Assuming non-adaptivity, the following assumptions are sufficient for our analysis below. Recall the notation that $A \in q \in Q_n$ means any leaf A of a tree structure q in the structure space Q_n. We make the following assumptions of the tree building process:

(L1) Asymptotic locality. Writing $diam(A) = sup_{x,y \in A}|x - y|$, we require

$$\sup_{A \in q \in Q_n} diam(A) = O(d_n), \quad d_n \to 0.$$

(L2) Minimal leaf size. If we write $V(\cdot)$ as the volume function in terms of Lebesgue measure, we require that

$$\inf_{A \in q \in Q_n} V(A) \geq O(v_n) > 0.$$

These assumptions together bound the space occupied by any leaf of any possible tree from being either too extensive or too small. It indicates that any leaf is a geometrically shrinking neighborhood of the points it contains, while the the number of neighborhood points increases. We will later specify the rates we require for Boulevard.

3.4.2 Fixed Design

We first consider a fixed sequence of samples with increasing sizes, i.e. for each n, the sample $(x_{n,1}, y_{n,1}), \ldots, (x_{n,n}, y_{n,n})$ is given. The first subscript n will be dropped when there is no ambiguity. We specify the rates for the size of leaf nodes as:

(R1) For some $\epsilon_1 > 0$,

$$d_n = O\left(n^{-\frac{1}{d+2}-\epsilon_1}\right).$$

(R2) For some $\epsilon_2 > \epsilon_1 > 0$,

$$\inf_{A \in q \in Q_n} \sum_{i=1}^{n} I(x_i \in A) \geq O\left(n^{\frac{2}{d+2}-d\epsilon_2}\right).$$

One compatible realization is

$$d_n = O\left(n^{-\frac{1}{d+1}}\right), \quad \inf_{A \in q \in Q_n} \sum_{i=1}^{n} I(x_i \in A) \geq O\left(n^{\frac{1}{d+2}}\right).$$

For simplicity all our proofs are under this setting. However, any other rates satisfying these conditions are also sufficient.

3.4.3 Missing Terminal Subsample

Starting here we use the abbreviations that

$$k_n^T = \mathbb{E}\left[s_n(x)\right], \quad K_n = \mathbb{E}\left[S_n\right], \quad r_n^T = k_n^T \left[\frac{1}{\lambda} I + K_n\right]^{-1}.$$

We take a close look at the missing terminal subsample issue due to which we can only guarantee $\|k_n\|_1 \leq 1$. Working with the tree construction rate as above, the subsample rate θ effectively determines how far $\|k_n\|_1$ is from 1.

Without loss of generality, let each terminal leaf contains no fewer than $n^{\frac{1}{d+2}}$ sample points before subsampling according to our assumed rates. If the subsample size is $\theta n = n^{\frac{d+1}{d+2}} \log n$, i.e. $\theta = n^{-\frac{1}{d+2}} \log n$, the chance of missing terminal subsample in a given leaf is

$$\begin{aligned} p(n,\theta) = \frac{\binom{n-n^{\frac{1}{d+2}}}{\theta n}}{\binom{n}{\theta n}} &= \frac{(n-\theta n)(n-\theta n-1)\cdots(n-\theta n-n^{\frac{1}{d+2}}+1)}{n(n-1)\cdots(n-n^{\frac{1}{d+2}}+1)} \\ &\leq \left(\frac{n-\theta n}{n-n^{\frac{1}{d+2}}}\right)^{n^{\frac{1}{d+2}}} = \left(\frac{1-n^{-\frac{1}{d+2}}\log n}{1-n^{-\frac{d+1}{d+2}}}\right)^{n^{\frac{1}{d+2}}} \\ &\leq e \cdot \left(1-n^{-\frac{1}{d+2}}\log n\right)^{n^{\frac{1}{d+2}}} \leq O\left(\frac{1}{n}\right). \end{aligned}$$

Therefore, for any x, $1 - \|k_n\|_1 \leq O\left(\frac{1}{n}\right)$ if we use subsample size at least of $n^{\frac{d+1}{d+2}} \log n$. This requires the subsample to be relatively large, which is compatible with, practically,

both constant subsample rate i.e. θ is constant, or $\log n$ subsample rate i.e. $\theta = (\log n)^{-1}$. We will refer to $p(n, \theta)$ as the *missing weight* in subsequent proofs.

To reach a similar statement for r_n, we first examine K_n since every row and column of K_n suffers from missing terminal subsample. The conclusion is summarized in the following lemma, whereas the detail calculations are in Appendix 3.7.3.

Lemma 3.6. *Using above settings and notations,*

$$\left| \sum_{i=1}^{n} r_{n,i} - \frac{\lambda}{1+\lambda} \right| \leq O\left(\frac{1}{n}\right).$$

3.4.4 Exponential Decay of Influence and Asymptotic Normality

The prediction that Boulevard makes at a point is a linear combination of responses $y_1, \ldots, y_n$ whose coefficients are given by r_n. Distant points ideally are less influential on the prediction, and such decay of influence in our case is exponential. To show this, we first introduce the notation of vector component selection. Given any n-vector v and an index set D, denote

$$v\big|_D = \begin{bmatrix} v_1 \cdot I(1 \in D) \\ \vdots \\ v_n \cdot I(n \in D) \end{bmatrix}.$$

Easy to verify that $v = v\big|_D + v\big|_{D^c}$.

Lemma 3.7. *Given sample $(x_1, y_1), \ldots, (x_n, y_n)$, a point of interest x, set $l_n = \frac{\log n}{-\log \lambda} =$*

$c_1 \log n$, *and define index set* $D_n = \{i : |x_i - x| \leq l_n \cdot \cdot d_n\}$, *then*

$$\left\| r_n \big|_{D_n^c} \right\|_1 \leq O\left(\frac{1}{n}\right).$$

Lemma 3.7 indicates that Boulevard trees will asymptotically rely on a $\log n$ shrinking neighborhood around the point of interest. Given sample size n and a point of interest x, we can therefore define $B_n = \left\{i \middle| |x_i - x| \leq d_n\right\}$ and $D_n = \left\{i \middle| |x_i - x| \leq l_n \cdot d_n\right\}$. B_n contains all points that have direct influence on x in a single tree, and D_n contains the points that dominate the prediction at x. $|B_n|$ and $|D_n|$ follow Binomial distributions with parameters depending on the local covariate density. These two quantities will appear in later proofs through the following lemma, whose proof results from simply verifying the Lindeberg-Feller condition for sums of Bernoulli random variables.

Lemma 3.8. *Assume* $X_1, \ldots, X_n, \ldots$, *independent binomial random variables s.t.* $X_i \sim$ *Binom*(n, p_n) *and* $np_n \to \infty$.

$$\frac{X_n - np_n}{\sqrt{np_n(1-p_n)}} \xrightarrow{d} N(0,1).$$

We are now ready to show the limiting distribution of fixed design cases. We check the Lindeberg-Feller condition for the sequence of predictions $\hat{f}_n(x)$ .The following lemma is used to bound $\|k_n\|$ and $\|r_n\|$.

Lemma 3.9. *With increasing n and sample* $(x_{n,1}, y_{n,1}), \ldots (x_{n,n}, y_{n,n})$ *at size n, assume* $|B_n| \geq O\left(n \cdot d_n^d\right)$ *and*

$$\inf_{A \in q \in Q_n} \sum_{i=1}^{n} I(x_{n,i} \in A) \geq O\left(n^{\frac{1}{d+2}}\right),$$

then

$$O\left(n^{-\frac{1}{2}\frac{1}{d+1}}\right) \leq \|k_n\|, \|r_n\| \leq O\left(n^{-\frac{1}{2}\frac{1}{d+2}}\right).$$

Theorem 3.10. *For given $x \in [0,1]^d$, suppose we have fixed sample $(x_{n,1}, y_{n,1}), \ldots, (x_{n,n}, y_{n,n})$ for each n s.t. $\left\|k_n^T\right\|_\infty \leq O\left(n^{-\frac{1}{d+2}}\right)$. Write $f(X_n) = (f(x_1), \ldots, f(x_n))^T$, then*

$$\frac{\hat{f}_n(x) - r_n^T f(X_n)}{\left\|r_n^T\right\|} \xrightarrow{d} N(0, \sigma_\epsilon^2).$$

Proof. Notice that

$$\hat{f}_n(x) - r_n^T f(X_n) = r_n^T \vec{\epsilon}_n.$$

To obtain a CLT we check the Lindeberg-Feller condition of $r_n^T \vec{\epsilon}_n$, i.e. for any $\delta > 0$,

$$\lim_n \frac{1}{\|r_n\|^2 \sigma_\epsilon^2} \sum_{i=1}^n \mathbb{E}\left[(r_{ni}\epsilon_i)^2 I(|r_{ni}\epsilon_i| > \delta \|r_n\| \sigma_\epsilon)\right] \to 0.$$

Since $\|k_n\|_\infty \leq O\left(n^{-\frac{1}{d+2}}\right)$ and $\left[\frac{1}{\lambda} I + K_n\right]^{-1}$ having row sums of $\frac{\lambda}{1+\lambda} + O\left(n^{-1}\right)$, we have

$$\|r_n\|_\infty \leq \|k_n\|_\infty \cdot \left\|\left[\frac{1}{\lambda} I + K_n\right]^{-1}\right\|_1 \leq O\left(n^{-\frac{1}{d+2}}\right).$$

Furthermore, since $\|r_n\| \geq O\left(n^{-\frac{1}{2}\frac{1}{d+1}}\right)$, we get

$$\frac{\|r_n\|_\infty}{\|r_n\|} \leq O\left(n^{-\frac{1}{d+2}+\frac{1}{2}\frac{1}{d+1}}\right),$$

which justifies the Lindeberg-Feller condition when ϵ is sub-Gaussian by

$$\begin{aligned}
\sum_{i=1}^n \mathbb{E}\left[(r_{ni}\epsilon_i)^2 I(|r_{ni}\epsilon_i| > \delta \|r_n\| \sigma_\epsilon)\right] &\leq \sum_{i=1}^n r_{ni}^2 \sqrt{\mathbb{E}\left[\epsilon_i^4\right] \cdot \mathbb{E}\left[I(|r_{ni}\epsilon_i| > \delta \|r_n\| \sigma_\epsilon)^2\right]} \\
&\leq \sum_{i=1}^n r_{ni}^2 \sqrt{\mathbb{E}\left[\epsilon_i^4\right]} \cdot \sqrt{P\left(|\epsilon_i| \geq \frac{\delta \|r_n\| \sigma_\epsilon}{r_{ni}}\right)}
\end{aligned}$$

$$\leq \sum_{i=1}^{n} r_{ni}^2 \sqrt{\mathbb{E}\left[\epsilon_i^4\right]} \sqrt{2\exp\left(-\frac{1}{2\sigma_\epsilon^2} \cdot \left(\frac{\delta \|r_n\| \sigma_\epsilon}{r_{ni}}\right)^2\right)}$$
$$\leq \|r_n\|^2 \exp\left(-O\left(n^{\frac{2}{d+2}-\frac{1}{d+1}}\right)\right) \longrightarrow 0,$$

since

$$P(\epsilon > t) \leq \exp\left(-\frac{t^2}{2\sigma_\epsilon^2}\right)$$

for sub-Gaussian ϵ. □

3.4.5 Random Design

In this section we analyze the random design case where the covariates $x_1, \ldots, x_n$ are considered randomly drawn from the underlying distribution. To extend the scope of the fixed design limiting distribution to the random design, we start from the following lemma.

Lemma 3.11. *Assume $X : \Omega_1 \to S$, independent of $\epsilon : \Omega_2 \to S$, $\{f_n : S \times S \to \mathbb{R}\}$ sequence of measurable functions. Assuming for a.s. $x \in \Omega_1$,*

$$f_n(x, \epsilon) \xrightarrow{d} N(0, 1).$$

Then

$$f_n(X, \epsilon) \xrightarrow{d} N(0, 1).$$

The idea behind the lemma is to incorporate the sample randomness by showing an almost sure point-wise convergence conclusion in a well-defined probability space. To translate the lemma into our context, we extend the original covariate and error space by

Kolmogorov's extension theorem. Define $(x_1, \dots) = \mathbf{X} \in [0,1]^{d\times\mathbb{N}}$ and $\epsilon = (\epsilon_1, \dots) \in \mathbb{R}^{\mathbb{N}}$, where the probability measures on $[0,1]^{d\times\mathbb{N}}$ and $\mathbb{R}^{\mathbb{N}}$ are uniquely decided by the product measures on the cylinder spaces reflecting i.i.d. sampling i.e. $y_i = f(x_i) + \epsilon_i$ for $i \in \mathbb{N}$. Write π_i the cumulative coordinate projection, i.e. $\pi_i(a_1, \dots, a_n, \dots) = (a_1, \dots, a_i)$. We can calculate k_n and K_n w.r.t. $\Pi_n = (\pi_n(\mathbf{X}), \pi_n(\epsilon))$. Thus

$$\rho_n(\mathbf{X}, \epsilon) = \frac{\hat{f}_n(x; \Pi_n) - k_n^T(x; \Pi_n)[\frac{1}{\lambda} I + K_n(\Pi_n)]^{-1} f(\Pi_n)}{\left\| k_n(x; \Pi_n)^T [\frac{1}{\lambda} I + K_n(\Pi_n)]^{-1} \right\|}$$

reflects the prediction after using a random sample of size n. Using Lemma 3.11, CLT of ρ_n requests an almost surely claim of Theorem 3.10 where the sequence of $(x_1, y_1), \dots, (x_n, y_n)$ comes from $(\pi_n(\mathbf{X}), \pi_n(\epsilon))$.

To help develop our analysis, we further increase the leaf size by a small amount assuming that the minimal terminal leaf geometric volume v_n follows

$$v_n = \frac{n^{\frac{1}{d+2}+\nu}}{n} = n^{-\frac{d+1}{d+2}+\nu} \leq n^{-\frac{d}{d+1}} = O\left(d_n^d\right)$$

for small $\nu > 0$. The following lemma shows the asymptotic normality where the mean depends on the random sample, whose proof is in Appendix 3.7.3.

Lemma 3.12. *For given $x \in [0,1]^d$, suppose we have random sample $(x_1, y_1), \dots, (x_n, y_n)$ for each n. If we restrict the cardinality of tree space Q_n by*

$$|Q_n| \leq O\left(\frac{1}{n} \exp\left(\frac{1}{2} n^{\frac{1}{d+2}}\right)\right),$$

then

$$\frac{\hat{f}_n(x) - r_n^T f(X_n)}{\left\| r_n^T \right\|} \xrightarrow{d} N(0, \sigma_\epsilon^2).$$

The proof of Lemma 3.12 also allows us to substitute all $O(\cdot)$ by $O_p(\cdot)$ in the analyses of random design. Further, we can replace the data driven mean $r_n^T f(X_n)$ by its population version $\frac{\lambda}{1+\lambda} f(x)$. Combining all above we obtain the main theorem of this chapter that the limiting distribution of the random design in our case is normal.

Theorem 3.13. *For given $x \in [0,1]^d$,*

$$\frac{\hat{f}_n(x) - \frac{\lambda}{1+\lambda} f(x)}{\left\| r_n^T \right\|} \xrightarrow{d} N(0, \sigma_\epsilon^2).$$

Proof. We first show that for given $x \in [0,1]^d$,

$$\frac{r_n^T f(X_n) - \frac{\lambda}{1+\lambda} f(x)}{\left\| r_n^T \right\|} \xrightarrow{p} 0.$$

Recall the index set $D_n = \{i : |x_i - x| \le l_n \cdot d_n\}$. Denote $\Delta = \frac{\lambda}{1+\lambda} - \sum_{i=1}^n r_{n,i} = O\left(n^{-1}\right)$ and $\tilde{f}(x) = (f(x), \ldots, f(x))^T$ an n-vector. We split

$$\begin{aligned}
\frac{r_n^T f(X_n) - \frac{\lambda}{1+\lambda} f(x)}{\left\| r_n^T \right\|} &= \frac{r_n^T [f(X_n) - \tilde{f}(x)]}{\left\| r_n^T \right\|} - \frac{\Delta \cdot f(x)}{\left\| r_n^T \right\|} \\
&= -\frac{\Delta \cdot f(x)}{\left\| r_n^T \right\|} + \frac{r_n\big|_{D_n} \cdot [f(X_n) - \tilde{f}(x)]\big|_{D_n}}{\|r_n\|} + \frac{r_n\big|_{D_n^c} \cdot [f(X_n) - \tilde{f}(x)]\big|_{D_n^c}}{\|r_n\|}.
\end{aligned}$$

By replacing $O(\cdot)$ in the fixed case by $O_p(\cdot)$ in the random design case, recall that

$$O_p\left(n^{-\frac{1}{2}\frac{1}{d+1}}\right) \le \|k_n\|, \|r_n\| \le O_p\left(n^{-\frac{1}{2}\frac{1}{d+2}}\right).$$

On one hand, we notice that

$$\left| r_n\big|_{D_n^c} \cdot [f(X_n) - \tilde{f}(x)]\big|_{D_n^c} \right| \le \left\| r_n\big|_{D_n^c} \right\|_1 \cdot \left\| [f(X_n) - \tilde{f}(x)]\big|_{D_n^c} \right\|_\infty \le O_p\left(\frac{1}{n} \cdot 2M_f\right) = O_p\left(n^{-1}\right).$$

Therefore

$$\frac{r_n\big|_{D_n^c} \cdot [f(X_n) - \tilde{f}(x)]\big|_{D_n^c}}{\|r_n\|} \xrightarrow{p} 0.$$

And similarly since $|\Delta| \leq O\left(n^{-1}\right)$,

$$\frac{\Delta \cdot f(x)}{\|r_n\|} \xrightarrow{p} 0.$$

On the other hand, we can show similarly as $|B_n|$ that $|D_n| = O\left(n \cdot (l_n \cdot d_n)^d)\right)$ a.s. and therefore

$$\begin{aligned}
\frac{\left|r_n\big|_{D_n} \cdot [f(X_n) - \tilde{f}(x)]\big|_{D_n}\right|}{\|r_n\|} &\leq \frac{\left\|r_n\big|_{D_n}\right\| \left\|[f(X_n) - \tilde{f}(x)]\big|_{D_n}\right\|}{\|r_n\|} \\
&\leq \left\|[f(X_n) - \tilde{f}(x)]\big|_{D_n}\right\| \\
&\leq O_p\left(\sqrt{n \cdot (l_n d_n)^d \cdot (l_n d_n \cdot \alpha)^2}\right) \\
&= O_p\left(\sqrt{n \cdot \log_n^{d+2} \cdot d_n^{d+2}}\right) \\
&= O_p\left(\sqrt{n \cdot \log_n^{d+2} \cdot n^{-\frac{d+2}{d+1}}}\right) \\
&= O_p\left((\log n)^{\frac{d+2}{2}} n^{-\frac{1}{2}\frac{1}{d+1}}\right).
\end{aligned}$$

Therefore

$$\frac{r_n\big|_{D_n} \cdot [f(X_n) - \tilde{f}(x)]\big|_{D_n}}{\|r_n\|} \xrightarrow{p} 0.$$

Combining the above calculations gives the result that

$$\frac{r_n^T f(X_n) - \frac{\lambda}{1+\lambda} f(x)}{\left\|r_n^T\right\|} \xrightarrow{p} 0.$$

Therefore by Slutsky's Theorem,

$$\frac{\hat{f}_n(x) - \frac{\lambda}{1+\lambda} f(x)}{\left\|r_n^T\right\|} = \frac{\hat{f}_n(x) - r_n^T f(X_n)}{\left\|r_n^T\right\|} + \frac{r_n^T f(X_n) - \frac{\lambda}{1+\lambda} f(x)}{\left\|r_n^T\right\|} \xrightarrow{d} N(0, \sigma_\epsilon^2).$$

□

Instead of the whole signal, Boulevard converges to $\frac{\lambda}{1+\lambda}$ of it. In standard boosting, we expect to converge to the whole signal. Boosting after this point will result in a random forest regressing on pure noise, which is redundant. In comparison, Boulevard down-weighs the boosting history to regularize that each tree in the finite ensemble reflects partial signal. It thus avoids being dominated by the first few trees then repeatedly fitting on noise. In practice, as we showed that the prediction from Boulevard is consistent w.r.t $\frac{\lambda}{1+\lambda} f(x)$, we simply rescale it by $\frac{1+\lambda}{\lambda}$ to retrieve the whole signal.

3.4.6 Undersmoothing, Tree Space Capacity and Subsampling

In the expression in Theorem 3.13, the mean is deterministic, but the variance is random. From results on kernel ridge regression, we would expect that this stochastic variance converges in probability if the random forest kernel behaves as generic kernel with a shrinking bandwidth. From a theoretical perspective, the optimal rate of $\left\| r_n^T \right\|$ is bounded from below by $O\left(n^{-\frac{1}{2}\frac{1}{d+1}}\right)$, which corresponds to the optimal nonparametric regression rate using $\frac{1}{2}$-Hölder continuous functions as base learners (Stone, 1982). In practice, $\left\| r_n^T \right\|$ relies on the specific method of growing the boosted trees, therefore may vary from case to case.

Furthermore, this demonstrates that with carefully structured trees the prediction is consistent while the variance involves no signal but the error. It acts like an undersmoothed local smoother whose bias term shrinks faster than the variance term.

We have a strict requirement that the tree terminal node size grows at a rate between $O\left(n^{\frac{1}{d+1}}\right)$ and $O\left(n^{\frac{1}{d+2}}\right)$ to guarantee undersmoothing. Any log term is allowed to be added

to the existing polynomial result without changing the behavior. We notice that different subsample rates (i.e. $\log n$ in Wager et al. (2014), $\sqrt{n}$ in Mentch and Hooker (2016)) have been applied for measuring uncertainty. In comparison, Boulevard algorithm requires a relatively restricted rate between these. In addition, though Boulevard training implements subsampling at each iteration, this does not influence the asymptotic distribution. The impact of subsampling is on the possible deviation from the mean process therefore the convergence speed if we assume non-adaptivity.

In the proof we have required the size of tree space to scale at a rate of $\frac{1}{n}\exp\left(\frac{1}{2}n^{\frac{1}{d+2}}\right)$. In comparison, Wager and Walther (2015) have shown that, in fixed dimension, any tree can be well approximated by a collection of $O(\exp(\log n)^2)$ hyper rectangles. Therefore the capacity of our designated tree space is decently large from a practical perspective.

3.5 Eventual Non-adaptivity

All the results mentioned above have assumed the non-adaptivity of the boosting procedure of Boulevard in order to separate the tree structure from the leaf values. In standard boosting however, it is conventional and reasonable to decide tree structures on the current gradients in order to better exploit the gap between the prediction and the signal. Such procedures are known for their tendency to overfit which can be relieved by subsampling. However, when seeking to extend our results to this case we lose the easy identifiability of a Boulevard convergence point since the tree structure distribution changes at each iteration. We therefore need more assumptions and further theoretical development to extend

the asymptotic normality to a more practical Boulevard algorithm that allows the current gradient to determine tree structure.

A first approach to this is to relax non-adaptivity to eventual non-adaptivity. We postulate a convergent sequence of predictions, indicating that underlying the tree spaces will be stabilized after boosting for sufficiently long time. Here we introduce the notation $\mathbb{E}\left[S_n(Y, \hat{Y})\right]$ where $Y = (y_1, \ldots, y_n)^T$ and $\hat{Y} = (\hat{f}(x_1), \ldots, \hat{f}(x_n))^T$ indicating the expected tree structure given the gradient of the loss between observed responses and current predictions. In regression this is $Y - \hat{Y}$, and we will take this form into the following discussion instead of a generic gradient expression.

It is also worth noticing we can also justify non-adaptivity asymptotically in contrast to pursue eventual non-adaptivity, . Consider building decision trees at a given rate without pruning. When sample size increases, the tree structure also gets more and more granular until n gets sufficiently large that the granular segmentation is very similar to the segmentation given by randomized trees without using the greedy building strategy. This understanding is also supported by the current practice of honest trees that partially diminish the influence of responses on the tree structure.

3.5.1 Local Homogeneity and Contraction Regions

We start with trees whose splits are based on the optimal Gini gain (Breiman et al., 1984). For $(x_1, z_1), \ldots, (x_n, z_n)$, the chosen split minimizes the impurity in the form of

$$\inf_{L,R} \quad \sum_{i \in L} (z_i - \bar{z}_L)^2 + \sum_{i \in R} (z_i - \bar{z}_R)^2, \tag{3.5}$$

where $L \subset \{1, \ldots, n\}, R = L^C$. Once the optimal split is unique, i.e. the optimum has a positive margin over the rest, we could allow a small change of all y's values without changing the split decision. This also holds true if the split is decided by a subsample instead. In terms of adaptive boosting, this observation demonstrates *local homogeneity* that, except a set $\Omega_0 \subset \mathbb{R}^n$ with Lebesgue measure 0 where $(z_1, \ldots, z_n)^T = Y - \hat{Y} \in \Omega_0$ has multiple optima for (3.5), we can segment $\mathbb{R}^n$, the space of possible $Y - \hat{Y}$, into subsets $\bigsqcup_{i=1}^{\alpha} C_i = \mathbb{R}^n \backslash \Omega_0$ s.t. $\mathbb{E}\left[S_n(Y, \hat{Y})\right] = \mathbb{E}\left[S_n(Y, \hat{Y}')\right]$ for $Y - \hat{Y}, Y - \hat{Y}' \in C_i$ the same subset.

Notice that Gini gain is insensitive to scaling, i.e. multiplying $(y_1, \ldots, y_n)$ by a nonzero factor. Therefore all C_i's are open double cones in $\mathbb{R}^n$.

Definition 3.2 (Contraction Region)**.** Given the sample $(x_1, y_1), \ldots (x_n, y_n)$. Write $Y = (y_1, \ldots, y_n)$ and current prediction $\hat{Y} = (\hat{y}_1, \ldots, \hat{y}_n)$. Following the above segmentation $\bigsqcup_{i=1}^{\alpha} C_i = \mathbb{R}^n \backslash \Omega_0$. We call C_i a contraction region if $Y^* \in C_i$ for the following Y^*

$$Y^* = \lambda \mathbb{E}\left[S_n(Y, \hat{Y})\right](Y - Y^*), \text{ i.e. } Y^* = \left[\frac{1}{\lambda} I + \mathbb{E}\left[S_n(Y, \hat{Y})\right]\right]^{-1} \mathbb{E}\left[S_n(Y, \hat{Y})\right] Y,$$

for any $Y - \hat{Y} \in C_i$, where $\mathbb{E}\left[S_n(Y, \hat{Y})\right]$ is the unique structural matrix in this region.

The intuition behind this definition is that, as long as a Boulevard process stays inside a contraction region, the subsequent tree structures will be conditionally independent of the

predicted values. Therefore the path becomes non-adaptive, collapsing to Y^*. To achieve this eventual non-adaptivity, we would like to know when a Boulevard path is permanently contained in a contraction region.

We should point out here that we have not shown the existence and the uniqueness of such contraction regions. Such an analysis would rely on the split proposing methods, the sample and the choice of λ.

3.5.2 Escaping the Contraction Region

In this section we explore possible approaches to restrict a Boulevard process inside a contraction region. Assuming the existence of contraction regions, we recall Theorem 3.3 which indicates that the Boulevard process has positive probability of not moving far from the fixed point. We formally state this as follows.

Theorem 3.14. *Denote $B(x, r)$ the open ball of radius r centered at x in $\mathbb{R}^n$. Suppose $C \subset \mathbb{R}$ a contraction region, $Y^* \in C$ the contraction point and $B(Y, 2r) \subset C$ for some $r > 0$. Write $\hat{Y}_b$ the Boulevard process. For sufficiently large t,*

$$P\left(\hat{Y}_b \in C, \forall b \geq t \middle| \hat{Y}_t \in B(Y^*, r)\right) \longrightarrow 1, t \to \infty.$$

Proof. We refer to Theorem 3.3. Choose $\delta = r$, and choose T s.t. $\forall t > T$,

$$\frac{\lambda}{t}\left(1 + \sqrt{n}\right) 2\sqrt{n}M \leq \frac{r}{\sqrt{d}}, \text{ i.e. } \sup \|\epsilon_t\| \leq \frac{r}{\sqrt{d}},$$

In this case, $\beta = \left\|\hat{Y}_t\right\| + \delta - \sqrt{d}\sup_{t\geq T}\|\epsilon_t\| \geq \delta = r$. By the conditional independence of $\hat{Y}_t$ and $\epsilon_b, b > t$ in the contraction region,

$$\begin{aligned} P\left(\hat{Y}_b \in C, \forall b \geq t \middle| \hat{Y}_t \in B(Y^*, r)\right) &\geq P\left(\sup_{b>t}\left\|\hat{Y}_b - Y^*\right\| \leq \left\|\hat{Y}_t - Y^*\right\| + \delta \middle| \hat{Y}_t \in B(Y^*, r)\right) \\ &= P\left(\sup_{b>t}\left\|\hat{Y}_b - Y^*\right\| \leq \left\|\hat{Y}_t - Y^*\right\| + \delta\right) \\ &\geq 1 - \frac{4\sqrt{d}\sum_{b=t+1}^{\infty}\mathbb{E}\left[\epsilon_b^2\right]}{r^2} \longrightarrow 1. \end{aligned}$$

□

Theorem 3.14 guarantees neither the existence or the uniqueness of the contraction region. A possible *ad hoc* solution to the existence is to apply a *tail snapshot* which uses the tree space that applies to some iteration b^* for the rest of the boosting steps when the Boulevard path begins to become stationary. This manually enforces the conditional independence between tree structures and boosting gradients, leading to non-adaptivity after b^*. An example of Boulevard regression implementing the tail snapshot is detailed in Algorithm 3.4.

Algorithm 3.4 (Tail Snapshot Boulevard)**.**

- *Start with* $\hat{f}_0 = 0$.
- *For* $b = 1, \ldots,$ *given* $\hat{f}_b$*, calculate the gradient*

$$z_i \triangleq -\frac{\partial}{\partial u_i}\sum_{i=1}^{n}\frac{1}{2}\left(u_i - y_i\right)^2\Big|_{u_i=\Gamma_M(\hat{f}_b(x_i))} = y_i - \Gamma_M(\hat{f}_b(x_i));$$

- *If b^* is not found, update by $1 > \lambda > 0$ and the tree structure space Q_b decided by all subsamples of current gradient,*

$$\hat{f}_{b+1}(x) = \frac{b}{b+1}\hat{f}_b(x) + \frac{\lambda}{b+1}s_b(x; Q_b)(z_1, \ldots, z_n)^T,$$

where $s_b(x; Q)$ denotes the random tree structure vector based on tree space Q. If b^* is found, update by Q_{b^*} instead, i.e.

$$\hat{f}_{b+1}(x) = \frac{b}{b+1}\hat{f}_b(x) + \frac{\lambda}{b+1}s_b(x; Q_b^*)(z_1, \ldots, z_n)^T.$$

- *When b^* is not found, check the empirical training loss as a measure of the distance to the fixed point.*

$$L_{b+1} = \frac{1}{2n}\sum_{i=1}^{n}\left(\frac{\lambda}{1+\lambda}y_i - \hat{f}_{b+1}(x_i)\right)^2.$$

If $L_{b+1} < L^*$ a preset threshold, we claim Boulevard is close enough to a fixed point and choose the current $b+1$ to be b^*.

3.6 Empirical Study

We have conducted a minimalist empirical study to demonstrate the performance of Boulevard. Despite the fact that our purpose in developing Boulevard lies in statistical inference, we require its accuracy to be on par with other predominant tree ensembles, which is assessed on both simulated and real world data. In addition, we inspect the empirical limiting behavior of non-adaptive Boulevard to show its agreement with our theory.

3.6.1 Predictive Accuracy

We first compare Boulevard predictive accuracy with the following tree ensembles: Random Forest (RF), gradient boosted trees without subsampling (GBDT), stochastic gradient boosted trees (SGBDT), non-adaptive Boulevard achieved by completely randomized trees (rBLV), adaptive Boulevard whose tree structures are influenced by gradient values (BLV). All the tree ensembles build same depth of trees throughout the experiment.

Results on simulated data are shown in Figure 3.1. We choose sample size of 5000 and use the following two settings as underlying response functions: (1) $y = x_1 + 3x_2 + x_3x_4$ (top), and (2) $y = x_1 + 3x_2 + (1 - x_3)^2 + x_4x_5 + (1 - x_6)^6 + x_7$ (bottom). Error terms are Unif[-1,1] (left) and equal point mass on $\{-1, 1\}$ (right). Training errors are evaluated on the training set with noisy responses, while testing errors are evaluated using the truth from the underlying signal on a separate test set, which is why testing errors appear to be smaller than training errors. BLV and rBLV are comparable with RF, while all the three equal-weight tree ensembles are slightly inferior to GBM and SGBM.

Results on four real world data sets selected from UCI Machine Learning Repository (Dheeru and Karra Taniskidou, 2017; Tüfekci, 2014; Kaya et al., 2012) are shown in Figure 3.2. All curves are averages after 5-fold cross validation. Different parameters are used for different data sets. Rankings of the five methods in comparison are quite volatile here, nevertheless rBLV and BLV manage to achieve decent performance on test sets despite the fact that BLV has the lowest training error which is a common indicator for overfitting.

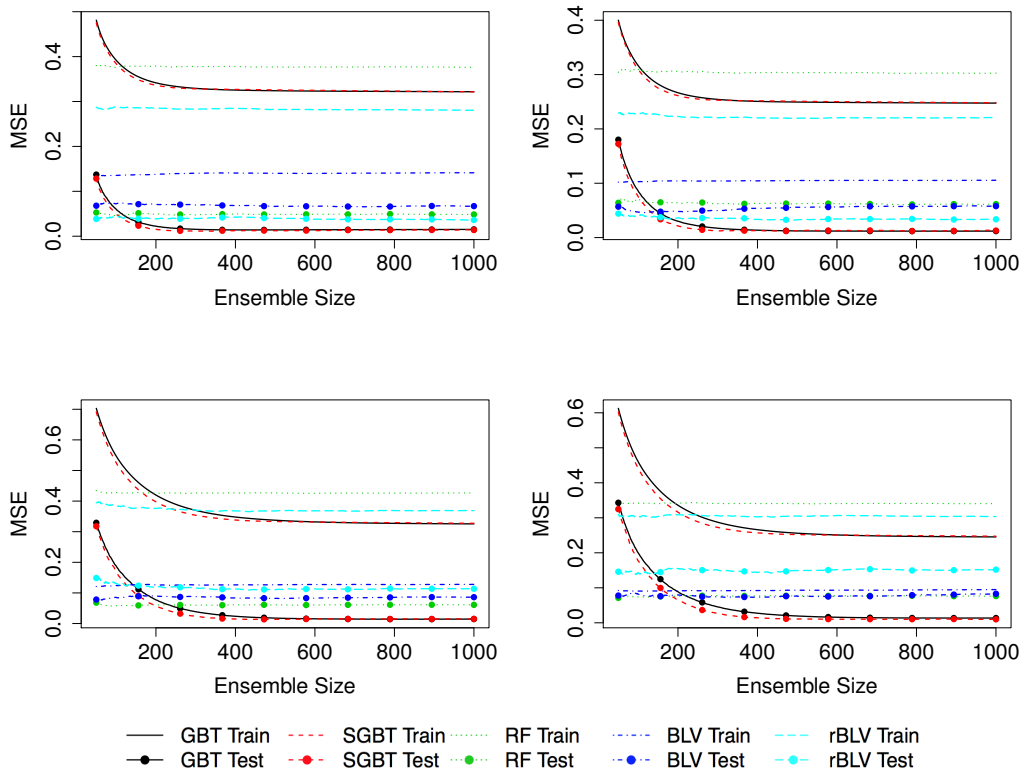


Figure 3.1: Training and testing error curves of tree ensembles on silmulated data.

3.6.2 Limiting Distribution

To examine the limiting behavior of non-adaptive Boulevard, we start with the model

$$y = x_1 + 3x_2 + x_3^2 + 2x_4x_5. \tag{3.6}$$

A set of 10 fixed test points are used along the experiments. We set a sample size of 1000, add different sub-Gaussian error terms to this signal and built non-adaptive Boulevard until ensemble size reaches 2000. This is repeated 1000 times with a new sample each

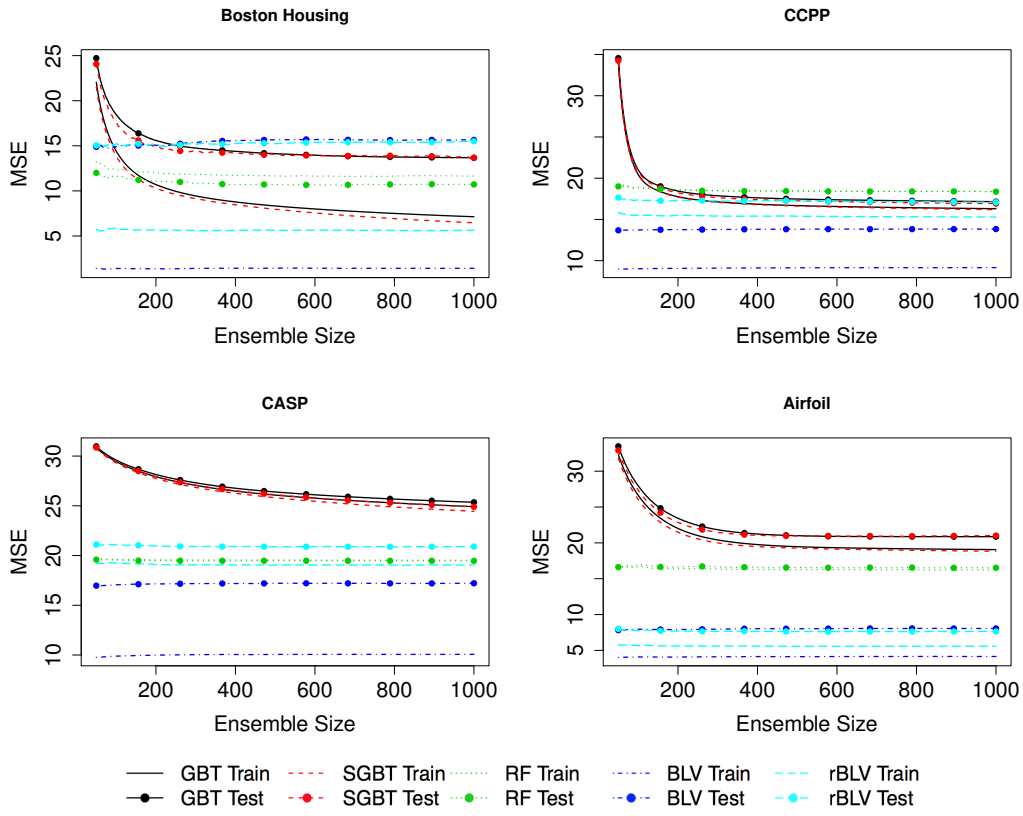


Figure 3.2: Training and testing error curves of tree ensembles on real world data sets.

time and we plot the distribution of the predictions in Figure 3.3. All these curves are undistinguishable from normal distribution by Kolmogorov-Smirnov test.

In addition, Table 3.1 shows the experiment in which we apply symmetric uniform errors and observe the scaling of prediction standard deviation along with the increase of error standard deviation.

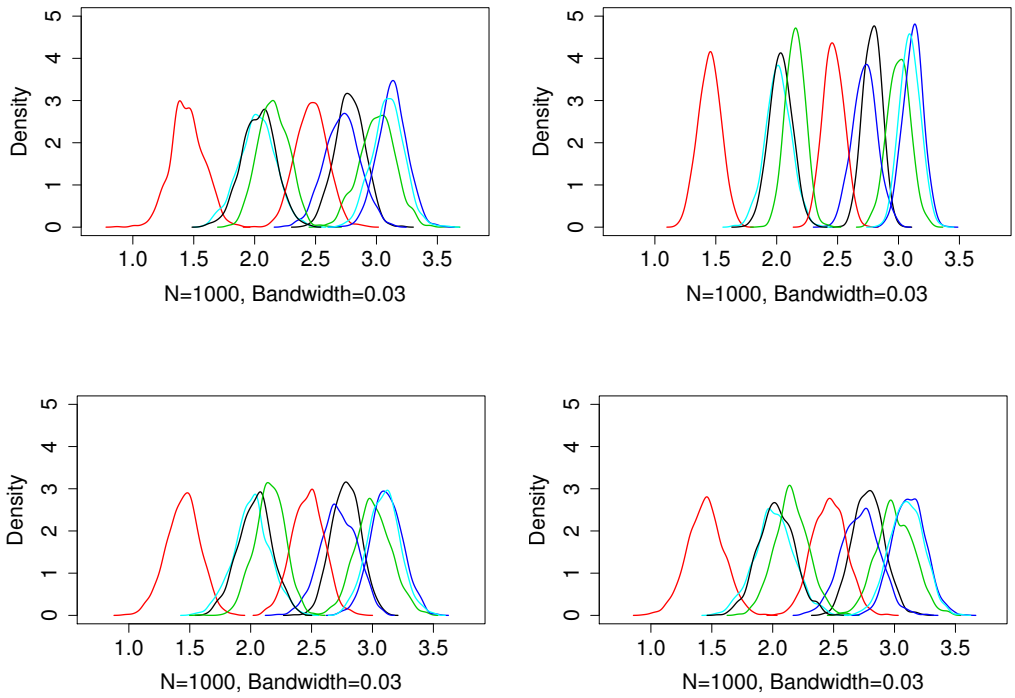


Figure 3.3: Distributions of predictions of test points with different error terms. The errors are N(0,1), Unif[-1,1], equal point mass at $\{-1, 1\}$, and half chance -1 half chance Unif[0,2], respectively.

3.6.3 Reproduction Interval

Similar to prediction intervals which quantify the uncertainty of future predictions, we introduce the *reproduction interval* as the uncertainty measure for where the prediction would be if it were made on another independent sample. Theorem 3.13 is used to create reproduction intervals for Boulevard. k_n in the stochastic variance is empirically estimated directly using the ensemble, while $[\frac{1}{\lambda}I + K_n]^{-1}$ is conservatively simplified to its largest possible norm λ. We then scale the variance estimate by 2 to account for having separate

Error\Fixed Point	1	2	3	4	5
0	0.030	0.044	0.044	0.049	0.050
Unif[-1,1]	0.067	0.089	0.096	0.087	0.096
Unif[-2,2]	0.119	0.154	0.172	0.158	0.162
Unif[-4,4]	0.243	0.271	0.278	0.278	0.288
Error\Fixed Point	6	7	8	9	10
0	0.037	0.038	0.033	0.032	0.040
Unif[-1,1]	0.083	0.081	0.074	0.071	0.082
Unif[-2,2]	0.152	0.122	0.139	0.137	0.145
Unif[-4,4]	0.317	0.284	0.289	0.318	0.254

Table 3.1: Prediction standard deviations scale with error standard deviations.

independent samples. We use the training sample to create reproduction intervals for the test points, then repeatedly train and predict each test point for another 100 times with a different sample each time. Figure 3.4 shows the 95% reproduction intervals we capture under different settings. We anticipate more accurate results with larger sample size.

Furthermore, we notice the uniform pattern of biases in those plots. This bias comes from two known causes. One is that we are using small samples which are far from guaranteeing the consistency. The other is because of the edge effects; the distance of the ten chosen test points to the center of the hypercube is respectively 0.000, 0.671, 0.894, 0.894, 0.894, 0.693, 0.520, 0.436, 0.510 and 0.469. We in general expect biased prediction when the point is near the boundary.

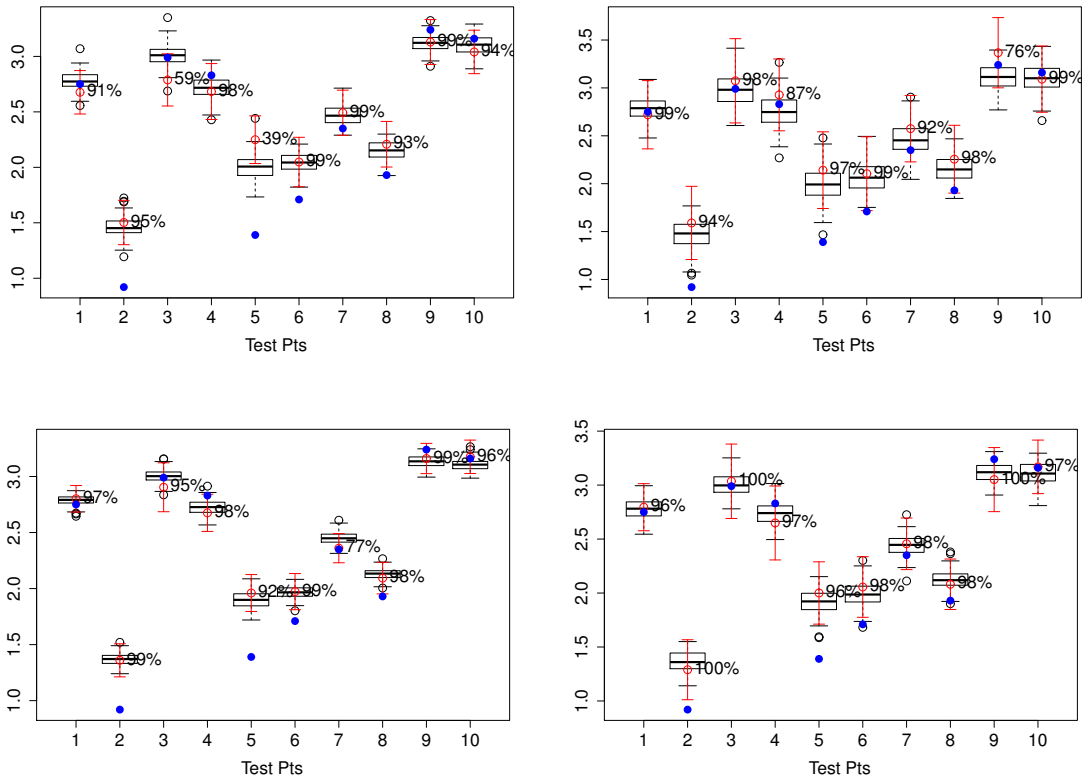


Figure 3.4: Reproduction intervals. Boxplots show distributions of predictions; red intervals are reproduction intervals; blue dots are truths. Sample sizes are 1000 (top row) and 5000 (bottom row), error terms Unif[-1,1] (left column) and Unif[-2, 2] (right column). Coverage is shown by numbers next to interval centers.

3.7 Proofs

In this section we list the complete proofs of all theorems covered above.

3.7.1 Properties of Tree Structure Matrices

Proof to Theorem 3.1

Proof. To prove (1), element-wise non-negativity is trivial. To show symmetry, consider any given $i \neq j$ and assume $x_i \in A$ and $x_j \in A'$ under the assumption of subsample uniformity,

$$\mathbb{E}_w [S_n]_{i,j} = \mathbb{E}_w \left[s_{n,j}(x_i) \right] = \frac{1}{\binom{n}{\theta n}} \sum_w \frac{I(x_j \in A) I(j \in w)}{\sum_{x_l \in A} I(l \in w)}$$
$$\mathbb{E}_w [S_n]_{j,i} = \mathbb{E}_w \left[s_{n,i}(x_j) \right] = \frac{1}{\binom{n}{\theta n}} \sum_w \frac{I(x_i \in A') I(i \in w)}{\sum_{x_l \in A'} I(l \in w)}$$

Therefore $\mathbb{E}_w [[] S_n]_{i,j} = \mathbb{E}_w [S_n]_{j,i} = 0$ if $A \neq A'$.

In the cases of $A = A'$, $I(x_j \in A) = I(x_i \in A') = 1$. We consider the following possibilities of w.

(a) For $i \notin w$, $j \notin w$,

$$\frac{I(j \in w)}{\sum_{x_l \in A} I(l \in w)} = \frac{I(i \in w)}{\sum_{x_l \in A} I(l \in w)} = 0.$$

(b) For $i \in w$, $j \in w$,

$$\frac{I(j \in w)}{\sum_{x_l \in A} I(l \in w)} = \frac{I(i \in w)}{\sum_{x_l \in A} I(l \in w)} = \frac{1}{\sum_{x_l \in A} I(l \in w)}.$$

(c) For $i \in w$, $j \notin w$, consider $w' = w \backslash \{i\} \cup \{j\}$ s.t. $\sum_{x_l \in A} I(l \in w) = \sum_{x_l \in A} I(l \in w')$,

$$\frac{I(j \in w')}{\sum_{x_l \in A} I(l \in w')} = \frac{I(i \in w)}{\sum_{x_l \in A} I(l \in w)} = \frac{1}{\sum_{x_l \in A} I(l \in w)}.$$

(d) Similarly, for $i \notin w$, $j \in w$, consider $w' = w \backslash \{j\} \cup \{i\}$,

$$\frac{I(j \in w)}{\sum_{x_l \in A} I(l \in w)} = \frac{I(i \in w')}{\sum_{x_l \in A} I(l \in w')} = \frac{1}{\sum_{x_l \in A} I(l \in w)}.$$

Since all w's are equally likely, we conclude by symmetry that $\mathbb{E}_w[S_n]_{i,j} = \mathbb{E}_w[S_n]_{j,i}$, hence $\mathbb{E}_w[S_n]$ is symmetric.

To prove (2), notice $\forall x_i, x_j, x_k \in A$,

$$\mathbb{E}_w[S_n]_{k,i} = \frac{1}{\binom{n}{\theta n}} \sum_w \frac{I(i \in w)}{\sum_{x_l \in A} I(l \in w)} = \mathbb{E}_w[S_n]_{j,i}.$$

Therefore $\mathbb{E}_w[S_n]$, after proper permutation to gather points in same leaves together, is diagonally blocked with equal entries in each diagonal block and 0 elsewhere, thus positive semi-definite.

To show (3), notice that S_n has row sums of ≤ 1 (not exactly 1 due to cases of missing subsample points in the leaf), so does $\mathbb{E}_w[S_n]$. Thus $\|\mathbb{E}_w[S_n]\|_1 \leq 1$. Similarly, $\mathbb{E}_w[S_n]$ has column sums of ≤ 1 due to symmetry and $\|\mathbb{E}_w[S_n]\|_\infty \leq 1$. By the Hlder inequality,

$$\rho(\mathbb{E}_w[S_n]) = \|\mathbb{E}_w[S_n]\| \leq \sqrt{\|\mathbb{E}_w[S_n]\|_1 \|\mathbb{E}_w[S_n]\|_\infty} \leq 1.$$

□

3.7.2 Stochastic Contraction

Definition 3.3 (Stochastic Contraction)**.** Given real-valued stochastic process $\{X_t\}_{t \in \mathbb{N}}$, a sequence of $0 < \lambda_t \leq 1$, define

$$\mathcal{F}_0 = \emptyset, \mathcal{F}_t = \sigma(X_1, \ldots, X_t),$$

$$\epsilon_t = X_t - \mathbb{E}[X_t | \mathcal{F}_{t-1}].$$

We call X_t a stochastic contraction if the following is satisfied

- Vanishing coefficients

$$\sum_{t=1}^{\infty}(1-\lambda_t)=\infty, \text{ i.e. } \prod_{t=1}^{\infty}\lambda_t=0.$$

- Mean contraction

$$\lambda_t X_{t-1} I(X_{t-1}\le 0)\le \mathbb{E}\left[X_t|\mathcal{F}_{t-1}\right]\le \lambda_t X_{t-1} I(X_{t-1}\ge 0), a.s..$$

- Bounded deviation

$$\sup|\epsilon_t|\to 0, \quad \sum_{t=1}^{\infty}\mathbb{E}\left[\epsilon_t^2\right]\le\infty.$$

Lemma 3.15. *If $\{X_t\}_{t\in\mathbb{N}}$ is a stochastic contraction.*

- *Almost sure convergence*

$$X_t \xrightarrow{a.s.} 0.$$

- *Kolmogorov maximal inequality. For any T,δ s.t. $\beta=|X_T|+\delta-\sup_{t>T}|\epsilon_t|>0$,*

$$P\left(\sup_{t\ge T}|X_t|\le |X_T|+\delta\right)\ge 1-\frac{4\sum_{t=T+1}^{\infty}\mathbb{E}\left[\epsilon_t^2\right]}{\min\{\delta^2,\beta^2\}}.$$

Proof. Define the stopping time of sign changes

$$T_0=0, T_k=\inf\{t>T_{k-1}|X_{t-1}\le 0, X_t>0 \text{ or } X_{t-1}\ge 0, X_t<0\}.$$

We now look at every realized path and examine the segment of the process holding the same sign. W.o.l.g., suppose $X_t\ge 0$ for $T_k<t<T_{k+1}$. Easy to check

$$X_t=\mathbb{E}\left[X_t|\mathcal{F}_{t-1}\right]+\epsilon_t\le\lambda_t X_{t-1}+\epsilon_t\le X_{t-1}+\epsilon_t\le X_{T_k}+\sum_{s=T_k+1}^{t}\epsilon_s. \tag{3.7}$$

Therefore $|X_t| \leq \left|X_{T_k}\right| + \left|\sum_{s=T_k+1}^{t} \epsilon_s\right|$, same for the negative case. Since ϵ_t's are independent and $\sum_{t=1}^{\infty} \mathbb{E}\left[\epsilon_t^2\right] \leq \infty$, $\sum_{t=1}^{\infty} \epsilon_t$ exists a.s.. Write $N = \sup_k\{T_k \leq \infty\}$ the number of sign changes.

If there are infinite sign changes, i.e. $N = \infty$, by sending $k \to \infty$, $\left|X_{T_k}\right| \xrightarrow{a.s.} 0$ and $\left|\sum_{s=T_k+1}^{T_k+n} \epsilon_s\right| \xrightarrow{a.s.} 0, \forall n > 0$. Hence $X_t \xrightarrow{a.s.} 0$.

If there are finite sign changes, we assume w.l.o.g. that for some k, $X_t \geq 0, \forall t \geq T_k$. (3.7) can be written as $X_t - \epsilon_t \leq X_{t-1}$ which indicates $X_t - \sum_{s=T_k+1}^{t} \epsilon_s$ is decreasing, therefore has a limit $(-\infty)$. Since $\sum_{s=T_k+1}^{\infty} \epsilon_s$ exists a.s., $X_t \xrightarrow{a.s.} c \geq 0$. Assume $c > 0$,

$$\sum_{s=T_k+1}^{\infty} \epsilon_s \geq \sum_{s=T_k+1}^{\infty} X_s - \lambda_s X_{s-1} = -\lambda_{T_k+1} X_{T_k} + \sum_{s=T_k+2}^{\infty} (1-\lambda_s) X_{s-1} = \infty,$$

which is a contradiction. Therefore $X_t \xrightarrow{a.s.} 0$.

To show the maximum inequality, we take the same notations above, and also look at segmentations by sign changes. For any t in the same segment as T,

$$|X_t| \leq |X_T| + \left|\sum_{s=T+1}^{t} \epsilon_t\right| \leq |X_T| + \sup_{T'>T} \left|\sum_{s=T+1}^{T'} \epsilon_s\right|.$$

For any t in a different segment starting at T',

$$|X_t| \leq \left|X'_T\right| + \left|\sum_{s=T'+1}^{t} \epsilon_t\right| \leq \left|X'_T\right| + \sup_{S>T'} \left|\sum_{s=T'+1}^{S} \epsilon_s\right| \leq \sup_{s>T} |\epsilon_s| + \left|\sum_{s=T'+1}^{S} \epsilon_s\right|.$$

Now we consider any possible sequence of $\{\epsilon_t, t > T\}$ and allow T', S to change. Kolmogorov maximal inequality implies

$$P\left(\sup_{i,j>T} \left|\sum_{s=i}^{j} \epsilon_s\right| \leq x\right) \geq P\left(\sup_{i>T} \left|\sum_{s=T}^{i} \epsilon_s\right| \leq \frac{x}{2}\right) \geq 1 - \frac{4\sum_{s=T}^{\infty} \mathbb{E}\left[\epsilon_s^2\right]}{x^2}.$$

The conclusion is obtained by noticing that $|X_t| \leq |X_T| + \delta$ for any $\{\epsilon_t\}_{t>T}$ satisfying

$$\sup_{i,j>T} \left| \sum_{s=i}^{j} \epsilon_s \right| \leq \min\{\delta, \beta\}.$$

□

Proof to Theorem 3.3

Proof. The idea is to define a sequence of adaptive orthonormal rotations $R_t \in \mathcal{F}_{t-1}$ to align the expected update with the previous step so that we can apply the $\mathbb{R}$ result component-wisely. Define $R_t \mathbb{E}[Z_t|\mathcal{F}_{t-1}] = \gamma_{t-1} Z_{t-1}$, for some $\gamma_{t-1} > 0, \gamma_{t-1} \in \mathcal{F}_{t-1}$. The contraction assumption also implies that $\gamma_{t-1} \leq \lambda_{t-1}$. Define a new process Z_i^* satisfying

1. $Z_1^* = Z_1, R_1 = I$,
2. writing $R_t^* = \prod_{i=1}^{n} R_i \in \mathcal{F}_{t-1}$ s.t. $Z_t^* = R_t^* Z_t = R_t^* \epsilon_t + R_t^* \mathbb{E}[Z_t|\mathcal{F}_{t-1}]$.

Above implies $\|Z_t\| = \left\|Z_t^*\right\|$, thus we need to prove the equivalence that $Z_t^* \xrightarrow{a.s.} 0$. Notice that Here $\sum_{i=1}^{n} R_i^* \epsilon_i$ is component-wisely a martingale with

$$\sum_{i=1}^{\infty} E[\left\|R_i^* \epsilon_i\right\|^2] = \sum_{i=1}^{\infty} E[\|\epsilon_i\|^2] < \infty,$$

hence $\sum_{i=1}^{n} R_i^* \epsilon_i$ exists a.s.. Since the construction aligns Z_t^* with $\mathbb{E}[Z_t^*|\mathcal{F}_{t-1}]$ we apply Lemma 3.15 to obtain almost sure convergence to 0 component-wisely, thus $\left\|Z_t^*\right\| \xrightarrow{a.s.} 0$. □

Proof to Corollary 3.5

Proof. Expanding $\hat{f}(x)$ gives

$$\begin{aligned}
\hat{f}(x) &= \lim_{B\to\infty} \frac{1}{B}\sum_{b=1}^{B} s_b(x)(Y - \hat{Y}_b) \\
&= \lim_{B\to\infty} \frac{1}{B}\sum_{b=1}^{B} s_b(x)(Y - Y^* + Y^* - \hat{Y}_b) \\
&= \lim_{B\to\infty} \frac{1}{B}\sum_{b=1}^{B} s_b(x)(Y - Y^*) + \lim_{B\to\infty} \frac{1}{B}\sum_{b=1}^{B} s_b(x)(Y^* - \hat{Y}_b) \\
&= \mathbb{E}\left[s_b(x)\right](Y - Y^*) + 0 \\
&= \mathbb{E}\left[s_n(x)\right]\left[\frac{1}{\lambda}I + \mathbb{E}\left[S_n\right]\right]^{-1} Y.
\end{aligned}$$

□

3.7.3 Asymptotic Normality

Proof to Lemma 3.6

Proof. Consider the expansion

$$\left[\frac{1}{\lambda}I + K_n\right]^{-1} = \lambda\sum_{i=0}^{\infty}\left((\lambda)^{2i}K_n^{2i} - (\lambda)^{2i+1}K_n^{2i+1}\right).$$

We examine the column sums of each of the matrix powers. Start with K_n^2,

$$\sum_{i=1}(K_n^2)_{i,1} = \sum_{i=1}^{n}\sum_{j=1}^{n}(K_n)_{i,j}(K_n)_{j,1} = \sum_{j=1}^{n}(K_n)_{j,1}\sum_{i=1}^{n}(K_n)_{i,j}.$$

Since K_n consists of structure vectors of sample points, for some $c > 0$,

$$1 - \frac{c}{n} \leq \sum_{j=1}^{n} (K_n)_{i,j} = \sum_{j=1}^{n} (K_n)_{i,j} \leq 1, \quad i = 1, \ldots, n.$$

Given K_n is nonnegative,

$$\left(1 - \frac{c}{n}\right)^2 \leq \sum_{i=1} (K_n^2)_{i,1} = \sum_{j=1}^{n} (K_n)_{j,1} \sum_{i=1}^{n} (K_n)_{i,j} \leq 1.$$

Repeating the same discussion yields

$$\left(1 - \frac{c}{n}\right)^m \leq \sum_{i=1} (K_n^m)_{i,1} \leq 1.$$

Therefore,

$$\begin{aligned}
\lambda\left(\frac{1}{1-\lambda^2(1-\frac{c}{n})^2} - \frac{\lambda}{1-\lambda^2}\right) &\leq \sum_{j=1}^{n} \left[\frac{1}{\lambda} I + K_n\right]^{-1}_{j,1} \\
&= \lambda\left(\sum_{i=0}^{\infty} (\lambda)^{2i} (K_n^{2i})_{j,1} - (\lambda)^{2i+1} (K_n^{2i+1})_{j,1}\right) \\
&\leq \lambda\left(\frac{1}{1-\lambda^2} - \frac{\lambda}{1-\lambda^2(1-\frac{c}{n})^2}\right),
\end{aligned}$$

where both the LHS and RHS reduce to $\frac{\lambda}{1+\lambda} + O\left(\frac{1}{n}\right)$. So is true for any column sum of $\left[\frac{1}{\lambda} I + K_n\right]^{-1}$. Now given k_n is nonnegative and $1 - \|k_n\|_1 \leq O\left(\frac{1}{n}\right)$ we reach the assertion. □

Proof to Lemma 3.7

Proof. Under locality, $k_{nj} = 0$ if $|x_i - x_j| > d_n$, while $[K_n]_{i,j} = 0$ if $|x_i - x_j| > d_n$. Recursively, if $|x_i - x_j| > l_n \cdot d_n$ then $[K_n^l]_{i,j} = 0$ for $l \leq l_n$. As k_n and K_n are element-wisely nonnegative, we again expand the matrix inverse

$$\left\| r_n \big|_{D_n^c} \right\|_1 = \sum_{|x - x_i| > l_n \cdot d_n} |r_{ni}| = \sum_{|x - x_i| > l_n \cdot d_n} \left| \sum_j k_{nj} \left[\frac{1}{\lambda} I + K_n\right]^{-1}_{j,i} \right|$$

$$
\begin{aligned}
&= \sum_{|x-x_i|>l_n\cdot d_n} \left| \sum_{|x-x_j|\le d_n} k_{nj} \left[\frac{1}{\lambda} I + K_n\right]^{-1}_{j,i} \right| \\
&\le \sum_{|x-x_j|\le d_n} k_{nj} \sum_{|x-x_i|>l_n\cdot d_n} \left| \left[\frac{1}{\lambda} I + K_n\right]^{-1}_{j,i} \right| \\
&\le \sum_{|x-x_j|\le d_n} k_{nj} \sum_{|x_i-x_j|>(l_n-1)\cdot d_n} \left| \left[\frac{1}{\lambda} I + K_n\right]^{-1}_{j,i} \right| \\
&\le \sum_{|x-x_j|\le d_n} k_{nj} \sum_{|x_i-x_j|>(l_n-1)\cdot d_n} \lambda \sum_{l=l_n}^{\infty} \lambda^l [K_n^l]_{j,i} \\
&\le \sum_{|x-x_j|\le d_n} k_{nj} \sum_{l=l_n}^{\infty} \lambda^{l+1} \\
&\le \sum_{l=l_n}^{\infty} \lambda^{l+1} = \frac{\lambda}{1-\lambda} \frac{1}{n}.
\end{aligned}
$$

□

Proof to Lemma 3.9

Proof. The idea is to bound k_{nj} from both above and below. The condition

$$
\inf_{A \in q \in Q_n} \sum_{i=1}^{n} I(x_i \in A) \ge O\left(n^{\frac{1}{d+2}}\right)
$$

implies that $k_{nj} \le O\left(n^{-\frac{1}{d+2}}\right)$. Given $||k_n||_1 \le 1$,

$$
||k_n|| \le \sqrt{||k_n||_1 \, ||k_n||_\infty} \le O\left(n^{-\frac{1}{2}\frac{1}{d+2}}\right)
$$

On the other hand, given $|B_n| \ge O\left(n \cdot d_n^d\right)$, there are at most

$$
O\left(n \cdot d_n^d\right) = O\left(n^{\frac{1}{d+1}}\right)
$$

k_{nj}'s that are positive. Since $\|k_n\|_1 \geq 1 - O(n^{-1})$,

$$\|k_n\| \geq O\left(\sqrt{\left(n^{-\frac{1}{d+1}}\right)^2 \cdot n^{\frac{1}{d+1}}}\right) = O\left(n^{-\frac{1}{2}\frac{1}{d+1}}\right).$$

Those bounds also work for $\|r_n\|$ given

$$\frac{\lambda}{1+\lambda} \leq eigen\left(\left[\frac{1}{\lambda}I + K_n\right]^{-1}\right) \leq \lambda.$$

□

Proof to Lemma 3.11

Proof. Probabilistic DCT guarantees that

$$\begin{aligned}
\lim_n P(f_n(X,\epsilon) \leq t) &= \lim_n \int\int \mathbb{1}_{\{f_n(x,\epsilon)\leq t\}} d\mu_x d\mu_\epsilon \\
&= \lim_n \int P(f_n(x,\epsilon) \leq t) d\mu_x \\
&= \int \lim_n P(f_n(x,\epsilon) \leq t) d\mu_x \\
&= \int \Phi(t) d\mu_x = \Phi(t).
\end{aligned}$$

□

Proof to Lemma 3.12

Proof. In order to prove the lemma, we combine Lemma 3.9, Theorem 3.10 and Lemma 3.11 and show that all assumptions are met from a point-wise perspective on $[0,1]^{d\times\mathbb{N}}$, i.e. fixed sample sequence are given by $\theta_n\mathbf{X}, n \geq 1$.

i) We show for a.s. $\mathbf{X}$, $\left|B_n^*\right| = |B_n(\theta_n \mathbf{X})| \geq O\left(n \cdot d_n^d\right)$. Consider random $\mathbf{X}$. Noticing $\mathbb{E}\left[|B_n|\right] = na_n = O\left(nd_n^d\right)$ and referring to CLT for binomials as $nd_n^d \to \infty$,

$$\frac{|B_n| - na_n}{\sqrt{na_n(1-a_n)}} \xrightarrow{d} N(0,1).$$

Take fixed $0 < c < 1$,

$$\begin{aligned} P\left(|B_n| \leq c \cdot na_n\right) \longrightarrow & \Phi\left(\frac{(c-1)na_n}{\sqrt{na_n(1-a_n)}}\right) \\ \leq & \Phi\left((c-1)\sqrt{na_n}\right) \\ \leq & O\left(\frac{1}{\sqrt{na_n}} \exp\left(-\frac{(c-1)^2 na_n}{2}\right)\right). \end{aligned}$$

Further, since $na_n = O\left(nd_n^d\right) = O\left(n^{\frac{1}{d+1}}\right)$,

$$\sum_{n=1}^{\infty} \frac{1}{\sqrt{na_n}} \exp\left(-\frac{(c-1)^2 na_n}{2}\right) \leq \infty.$$

As per Borel-Contelli, since

$$\sum_{n=1}^{\infty} P(|B_n(\theta_n \mathbf{X})| \leq c \cdot na_n) \leq \infty,$$

then for a.s. $\mathbf{X}$, events of $|B_n(\theta_n \mathbf{X})| \leq c \cdot na_n$ happens finite times. Since a_n is uniformly bounded away from 0 due to $\mu(x)$ is bounded, we reach our conclusion.

ii) To show

$$\inf_{A \in q \in Q_n} \sum_{i=1}^{n} I(x_i \in A) \geq O\left(n^{\frac{1}{d+2}}\right)$$

for a.s. $\mathbf{X}$, evaluate the CLT of binomial again

$$
\begin{aligned}
&P\left(\exists A \in q \in Q_n \text{ s.t. } \sum_{i=1}^{n} I(x_i \in A) \leq n^{\frac{1}{d+2}}\right) \\
\leq & O\left(|Q_n| \cdot |q| \cdot P\left(\sum_{i=1}^{n} I(x_i \in A) \leq n^{\frac{1}{d+2}}\right)\right) \\
\leq & O\left(|Q_n| \cdot n^{\frac{d+1}{d+2}} \cdot \Phi\left(\frac{n^{\frac{1}{d+2}} - n^{\frac{1}{d+2}+\nu}}{\sqrt{n^{\frac{1}{d+2}+\nu}\left(1 - n^{-\frac{d+1}{d+2}+\nu}\right)}}\right)\right) \\
\leq & O\left(|Q_n| \cdot n^{\frac{d+1}{d+2}} \cdot \Phi\left(-n^{\frac{1}{2}\left(\frac{1}{d+2}+\nu\right)}\right)\right) \\
\leq & O\left(|Q_n| \cdot n^{\frac{d+1}{d+2}} \cdot n^{-\frac{1}{2}\left(\frac{1}{d+2}+\nu\right)} \exp\left(-\frac{1}{2} n^{\frac{1}{d+2}+\nu}\right)\right) \\
\leq & O\left(\exp\left(-\frac{1}{2} n^{\frac{1}{d+2}+\nu}\right)\right) \longrightarrow 0.
\end{aligned}
$$

Therefore, noticing that

$$
\sum_{n=1}^{\infty} \exp\left(-\frac{1}{2} n^{\frac{1}{d+2}+\nu}\right) = \sum_{n=1}^{\infty} n^{-\frac{n^{\frac{1}{d+2}+\nu}}{2\log n}} < \infty,
$$

the Borel-Cantelli theorem indicates our assertion. Hence, for a.s. $\mathbf{X}^*$, $\theta_n \mathbf{X}^*$ satisfies the assumptions in Theorem 3.10. □

CHAPTER 4

TREE BOOSTED VARYING COEFFICIENT MODELS AND THEIR ASYMPTOTICS

4.1 Combining Parametric Models with Boosting

In this chapter we study the amalgamation of gradient boosting, especially gradient boosted decision trees (GBDT or GBM: Friedman, 2001), and varying coefficient models (VCM: Hastie and Tibshirani, 1993). A varying coefficient model is a semi-parametric model with coefficients that change along with each input. Under a general statistical learning setting with a set of covariates and some response of interest, a VCM isolates part of those covariates as *effect modifiers* based on which model coefficients are determined through a few varying coefficient mappings. These coefficients then get joined with the remaining covariates to generate a parametric prediction. To elaborate, consider performing least square regression on $(X, Z, Y) \in \mathbb{R}^p \times \mathbb{A} \times \mathbb{R}, i = 1, \ldots, n$ where $X = (X^1, \ldots, X^p)$, X and Z are the covariates and Y the response. One VCM regression can take the form of

$$g(\mathbb{E}[Y|X, Z]) = \beta^0(Z) + \sum_{i=1}^{p} \beta^i(Z) X^i, \tag{4.1}$$

with the parametric part being a generalized linear model with the link function g. In this context we would like to refer to X as the *predictive covariates* and Z the *action covariates (effect modifiers)* which are drawn from $\mathbb{A}$ the *action space*. $\beta^i(\cdot) : \mathbb{A} \to \mathbb{R}, i = 0, 1, \ldots, p$ are, conventionally nonparametric, *varying coefficient mappings*. While (4.1) maintains the linear structure, due to the dependence of β on any given Z, the model belongs to

a more complicated and flexible model space rather than the corresponding generalized linear model.

Our proposed model, tree boosted VCM, utilizes ensembles of gradient boosted decision trees as the varying coefficient mappings β. To demonstrate, for each $\beta^i, i = 0, \ldots, p$, let

$$\beta^i(z) = \sum_{j=1}^{b} t_j^i(z),$$

an additive boosted tree ensemble of size b with each t_j^i a decision tree constructed sequentially through gradient boosting. We will postpone the details of model construction to Section 2. This strategy yields a model of

$$g(\mathbb{E}[Y|X, Z]) = \sum_{j=1}^{b} t_j^0(Z) + \sum_{i=1}^{p} \left(\sum_{j=1}^{b} t_j^i(Z) \right) X^i. \tag{4.2}$$

Introducing VCM aligns with our attempt to answer the rising concern about model intelligibility and transparency, around which there are two branches of methods. We can either apply *post hoc* methods such that state-of-the-art "black box" models are constructed before we grant them meanings through analyzing their results. There is a sizable literature on this topic, from the appearance of local methods (Ribeiro et al., 2016) to recent applications on neural nets (Zhang and Zhu, 2018), random forests (Mentch and Hooker, 2016; Basu et al., 2018) and complex model distillation (Lou et al., 2012, 2013; Tan et al., 2017). However, objectivity is one inevitable challenge of tying explanations to models, especially in the presence of plentiful universal local methods capable of dealing with most models. Any use of *post hoc* analysis may be subject to justify the chosen ex-

planatory method over the others, which is likely to add an additional explanation selection phase on top of the existing model selection.

On the other hand, another branch of methods attempts to build interpretability into model structures, meaning that models should be the integration of simple and intelligible building blocks that they become accountable by human inspection once trained. Examples of this range from simple models as generalized linear models and decision trees, to models that guarantee monotonicity (You et al., 2017; Chipman et al., 2016) or have identifiable components (Melis and Jaakkola, 2018). Although having the advantage of not requiring *post hoc* examination, in contrast to the aforementioned methods, self-explanatory models are restricted by their possible model complexity and flexibility, potentially limiting their accuracy. This lack of flexibility also implies that such a model, unless possessing a granular structure, may only provide global interpretation because all observations are reasoned via an identical procedure. Such behavior prevents us from zooming into a small region in the sample space.

Following this discussion, VCM belongs to the second category as long as the involved parametric models are intelligible. It is an instant generalization of parametric methods to allow the use of local coefficients, which leads to improvements in model complexity and accuracy, whereas the predictions are still produced through parametric relations between predictive covariates and coefficients. This combination demonstrates a feasible means to balance the trade-off between flexibility and intelligibility.

A great amount of research has been conducted to study the asymptotic properties of different VCMs when splines or kernel smoothers are implemented as the nonparametric

varying coefficient mappings. We refer the readers to Park et al. (2015) for a comprehensive review. In this chapter we intend to conclude similar results regarding the asymptotics of tree boosted VCM.

4.1.1 Models under VCM

Under the settings of (4.1), Hastie and Tibshirani (1993) pointed out that VCM is the generalization of generalized linear models, generalized additive models, and various other semi-parametric models with careful choices of the varying coefficient mapping β.

We would like to mention two special cases that have drawn our attention. One is the functional trees introduced in Gama (2004). A functional tree segments the action space into disjoint regions, after which a parametric model gets fitted within each region using sample points inside. Logistic regression trees, for which there is a sophisticated building algorithm (LOTUS: Chan and Loh, 2004), belong to such model family. Their prediction on (x_0, z_0) is

$$P(\hat{y}_0 = 1) = \sum_{i=1}^{K} \frac{1}{1 + e^{-x_0^T \beta_i}} \cdot I\left(z_0 \in A_i\right) = \frac{1}{1 + e^{-x_0^T \beta_k}},$$

provided $\mathbb{A} = \coprod_{i=1}^{K} A_i$ the tree segmentation, $z_0 \in A_k$ and $\beta_k = \beta(z), \forall z \in A_k$. The conventional approach to determine functional tree structure is to recursively enumerate through candidate splits and choose the one that reduces the training loss the most between before and after splitting. Despite of the guaranteed stepwise improvement, such greedy strategy has the side effect of being both time consuming and mathematically intractable.

Another case is the partially linear regression that assumes

$$Y = X^T\beta + f(Z) + \epsilon_Z, \quad \epsilon_Z \sim N(0, \sigma^2(Z)),$$

where β is a global linear coefficient (see Härdle et al., 2012). It is equivalent to a least square VCM with all varying coefficient mappings except the intercept being constant.

4.1.2 Trees and VCM

While popular choices of varying coefficient mappings are either splines or kernel smoothers, it is a natural transition to consider exercising decision trees (CART: Breiman et al., 1984) and decision tree ensembles to serve as these nonparametric mappings. Using trees enables us to work adaptively with any action space $\mathbb{A}$ compatible with decision tree splitting logic, for example an arbitrary high dimensional mixture of continuous and discrete quantities, whereas traditional methods require to craft model structures case by case depending on the given $\mathbb{A}$.

We start with the straightforward attempts to utilize a single decision tree as varying coefficient mappings (Buergin and Ritschard, 2017; Berger et al., 2017). Although having a simple form, these implementations are also subject to the instability caused by the greedy tree building algorithm. Moreover, the mathematical intractability of decision trees prevents these single-tree based varying coefficient mappings from provable optimality. This instead suggests implementations through tree ensembles of either random forests or gradient boosting. One example is to use the linear local forests introduced in Friedberg et al. (2018) that perform local linear regression with an honest random forest kernel, while

the predictive covariates X are reused as the action covariates Z. In terms of boosting methods, Wang and Hastie (2014) proposed the first tree boosted VCM algorithm. They reduced the empirical risk by boosting using functional trees to fit the residuals to improve model coefficients, resulting in models of

$$g(\mathbb{E}[Y|X,Z]) = \left(\sum_{j=1}^{b} t_j(Z)\right)^T (1,X),$$

where each t_j returns a $(p+1)$ dimensional response. However, building a functional tree ensemble requires the construction and comparison of massive amounts of submodels and the joint optimization of all coefficients. In contrast, we aim to perform gradient boosting down on the coefficient level to comply with the standard boosting framework in order to separate the coefficients and to make tree boosted VCM coherent with existing boosting theories.

In the following sections, we explore the feasibility and statistical properties of adopting generic gradient boosted decision trees to serve as the nonparametric varying coefficient mappings for VCM. In Section 2, we share the perspective of analyzing such models as *local gradient descent* which creates functional coefficients and optimizes using local information. We will prove the consistency of this method in Section 3 and present a few empirical study results in Section 4. Further discussions on potential variations of this method follow in Section 5.

4.2 Tree Boosted Varying Coefficient Models

4.2.1 Notations

We will use the following notations in our discussion. We use superscripts $0, \ldots, p$ to indicate individual components, i.e. $\beta = (\beta^0, \ldots, \beta^p)^T$, and subscripts $1, \ldots, n$ to indicate sample points or boosting iterations. For any X when there is no ambiguity we assume X contains the intercept column, i.e. $X = (1, X^1, \ldots, X^p)$, so that $X^T\beta = \beta^0 + \sum_{i=1}^{p} X^i\beta^i$ can be used to specify a linear regression.

4.2.2 Boosting Framework

We start by looking at a parametric generalized linear model with coefficients $\beta \in \mathbb{R}^{p+1}$ using gradient descent. Given sample $(x_1, z_1, y_1), \ldots, (x_n, z_n, y_n)$ and a loss function l, gradient descent minimizes the empirical risk to search for the optimal $\hat{\beta}^*$ as

$$L(\hat{\beta}) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, x_i^T\hat{\beta}), \quad \hat{\beta}^* = \arg\min_{\hat{\beta}} L(\hat{\beta}).$$

To improve an interim $\hat{\beta}$, we move it in the negative gradient direction

$$\Delta_{\hat{\beta}} = \nabla_\beta L = -\nabla_\beta \left(\frac{1}{n}\sum_{i=1}^{n} l(y_i, x_i^T\beta)\Big|_{\beta=\hat{\beta}} \right),$$

to obtain a new iteration $\hat{\beta}' = \hat{\beta} + \lambda\Delta_{\hat{\beta}}$ for a positive and small learning rate $\lambda \ll 1$.

In order to extend this setting to varying coefficient models, we instead consider β to be a mapping $\beta = \beta(z) : \mathbb{A} \to \mathbb{R}^{p+1}$ so that it will apply to the covariates based on their

values in the action space. Writing estimate of β by $\hat{\beta} : \mathbb{A} \to \mathbb{R}^{p+1}$, the empirical risk remains a similar form

$$L(\hat{\beta}) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, x_i^T \hat{\beta}(z)).$$

We perform the same gradient calculation as above, but only pointwisely for now. It produces the negative gradient direction at the point z_i

$$\Delta_\beta(z_i) = -\nabla_\beta l(y_i, x_i^T \beta)\Big|_{\beta=\hat{\beta}(z_i)}. \tag{4.3}$$

As a result, we get the functional improvement of $\hat{\beta}$ captured at each of the sample points, i.e. $(z_1, \Delta_\beta(z_1)), \ldots, (z_n, \Delta_\beta(z_n))$. This observation leads us to employ gradient descent in functional space, also known as boosting (Friedman, 2001). For any function family $\mathcal{T}$ capable of regressing $\Delta_\beta(z_1), \ldots, \Delta_\beta(z_n)$ on $z_1, \ldots, z_n$, the corresponding ordinary boosting framework works as follows.

Algorithm 4.1 (Boosting coefficients)**.**

(B1) Start with an initial guess of $\hat{\beta}_0(\cdot)$.

(B2) For each component $j = 0, \ldots, p$ *of* $\hat{\beta}_b, b \geq 0$*, we calculate the pseudo gradient at each point as*

$$\Delta_{\beta_i}^j = -\frac{\partial l(y_i, x_i^T \beta)}{\partial \beta^j}\Big|_{\beta=\hat{\beta}_b(z_i)},$$

for $i = 1, \ldots, n$.

(B3) For each j*, find a good fit* $t_{b+1}^j \in \mathcal{T} : \mathbb{A} \to \mathbb{R}$ *on* $(z_i, \Delta_{\beta_i}^j), i = 1, \ldots, n$.

(B4) Update $\hat{\beta}_b$ with learning rate $\lambda \ll 1$.

$$\hat{\beta}_{b+1}(\cdot) = \hat{\beta}_b(\cdot) + \lambda \begin{bmatrix} t^0_{b+1}(\cdot) \\ \vdots \\ t^p_{b+1}(\cdot) \end{bmatrix}.$$

When the spline method is implemented, $\mathcal{T}$ is closed under addition so that we will expect the result of (B4) to be expressed as a set of coefficients of basis functions for $\mathcal{T}$. On the other hand, when we apply decision trees in place of (B3):

(B3') For each j, build a decision tree $t^j_{b+1} : \mathbb{A} \to \mathbb{R}$ on $(z_i, \Delta^j_{\beta_i}), i = 1, \ldots, n$,

the resulting varying coefficient mapping will be an additive tree ensemble, whose model space varies based on the ensemble size. We will refer to this method as *tree boosted VCM*.

Notice that the strategy of building a decision tree in (B3') influences the properties of the obtained tree boosted VCM. Recall that the standard CART strategy executes as follows.

(D1) Start at the root node.

(D2) Given a node, numerate candidate splits and evaluate them using all $(z_i, \Delta^j_{\beta_i})$ such that z_i is contained in the node.

(D3) Split on the best candidate split.

(D4) Keep splitting until stopping rules are met to form terminal nodes.

(D5) Calculate fitted terminal values in each terminal node using all $(z_i, \Delta^j_{\beta_i})$ such that z_i is contained in the terminal node.

As mentioned, recent developments on decision trees also suggest alternative strategies that produce better theoretical guarantees. We may consider *subsampling* that generates a subset $w \subset \{1, \ldots, n\}$ and only uses sample points indexed by w in (D2).

(D2') Given a node, numerate candidate splits and evaluate them using all $(z_i, \Delta^j_{\beta_i})$ such that $i \in w$ and z_i is contained in the node.

We may also consider honest trees which avoids using the responses, in our case $\Delta^j_{\beta_i}$, twice during both deciding the tree structure and deciding terminal values. Similarly as Boulevard, we can use a version of completely random trees which chooses the splits using solely z_i without evaluating the splits by the responses $\Delta^j_{\beta_i}$ in place of steps (D2) and (D3).

(D2*) Given a node, choose a random split based on z_i's contained in the node.

4.2.3 Local Gradient Descent with Tree Kernels

Decision tree fits in (B3') generate local linear combinations of pseudo-gradients thanks to the grouping effect carried by decision tree terminal nodes. To elaborate from a generic viewpoint, for all tree building strategy we discussed above we can introduce a kernel

smoother $K : \mathbb{A} \times \mathbb{A} \to \mathbb{R}$ such that the estimated gradient at any new z is given by

$$\Delta_\beta(z) = -\sum_{i=1}^{n} \left(\nabla_\beta l(y_i, x_i^T \beta) \Big|_{\beta=\hat{\beta}(z_i)} \right) \cdot \frac{K(z, z_i)}{\sum_{j=1}^{n} K(z, z_j)}. \tag{4.4}$$

In other words, with a fast decaying K, (4.4) can estimate the gradient at z locally using weights given by

$$S(z, z_i) = \frac{K(z, z_i)}{\sum_{j=1}^{n} K(z, z_j)}.$$

We would like to define such method as *local gradient descent*.

During standard tree boosting employing CART strategy, after a decision tree is constructed each iteration, its induced smoother K assigns equal weights to all sample points in the same terminal node. If we write $A(z_i) \subset \mathbb{A}$ the region in the action space corresponding to the terminal node containing z_i, we have $K(z, z_i) = I(z \in A(z_i))$ and we define the following

$$\mathbf{K}(z, z_i) \triangleq S(z, z_i) = \frac{I(z \in A(z_i))}{\sum_{j=1}^{n} I(z_j \in A(z_i))}$$

to be the *tree structure function* mentioned before where we also use the convention that $0/0 = 0$. The denominator is the size of z_i's terminal node and is equal to $\sum_{j=1}^{n} I(z_j \in A(z))$ when z and z_i fall in the same terminal node. In the cases where subsampling or completely random trees are employed for the purpose of variance reduction, $\mathbf{K}$ will be taken to be the expectation such that

$$\mathbf{K}(z, z_i) \triangleq \mathbb{E}[S(z, z_i)] = \mathbb{E}\left[\frac{I(z \in A(z_i)) I(i \in w)}{\sum_{j=1}^{n} I(z_j \in A(z_i)) I(i \in w) I(j \in w)} \right].$$

This expectation is taken over all possible tree structures and, if subsampling is applied, all possible subsamples w of a fixed size, and the denominator in the expectation is again the size of z_i's terminal node.

In particular, by carefully choosing the rates for tree construction, this tree structure function is related to the random forest kernel introduced in Scornet (2016) that takes the expectation of the numerator and the denominator separately as

$$\mathbf{K}_{RF}(z, z_i) = \frac{\mathbb{E}\left[I\left(z \in A(z_i)\right)\right]}{\mathbb{E}\left[\sum_{j=1}^{n} I\left(z_j \in A(z_i)\right)\right]},$$

in the sense that the deviations from these expectations are mutually bounded by constants.

Gradient boosting applied under nonparametric regression setting has to be accompanied by regularization such as using a complexity penalty or early stopping to prevent overfitting. When decision trees are implemented as the base learners, the complexity penalty is implicitly embedded in the tree parameters such as tree depth and terminal node size, while early stopping can be enforced during training. In fact, while we keep the parametric linear structure in VCM, local neighborhood weighting used for fitting the nonparametric coefficient mappings still adds to the model complexity. Therefore moderate restrictions, especially growth rates, have to be applied to avoid building saturated models with respect to the action space.

4.2.4 Examples

Tree boosted VCM generates a two-phase model such that the varying coefficient mappings generate effect modifiers and these effect modifiers join with predictive covariates linearly. In order to understand the varying coefficient mappings on the actions space, we

provide a visualized example here by considering the following data generating process:

$$X \sim \text{Unif}[0,1]^3, Z = (Z^1, Z^2) \sim \text{Unif}[0,1]^2, \epsilon \sim N(0, 0.25),$$

$$Y = X^T \begin{bmatrix} 0 \\ 3 \\ -5 \end{bmatrix} \cdot I(Z^1 + Z^2 < 1) + X^T \begin{bmatrix} -5 \\ 10 \\ 0 \end{bmatrix} \cdot I(Z^1 + Z^2 \geq 1) + \epsilon.$$

We generate a sample of size 1,000 from the above distribution, apply the tree boosted VCM with 400 trees, and obtain the following estimation of the varying coefficient mappings β on Z in Figure 4.1. Our fitted values accurately capture the true coefficients.

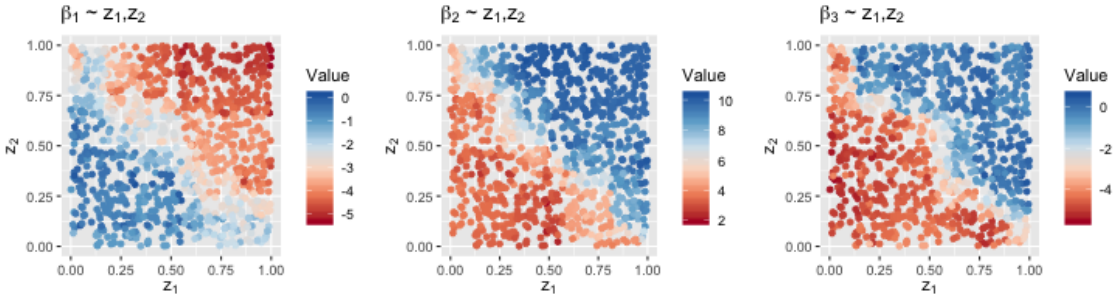


Figure 4.1: Example of varying coefficient mappings on the action space under the OLS settings.

Switching to logistic regression setting and assuming similarly that

$$\text{logit}P(Y = 1) = X^T \begin{bmatrix} 0 \\ 3 \\ -5 \end{bmatrix} \cdot I(Z^1 + Z^2 < 1) + X^T \begin{bmatrix} -5 \\ 10 \\ 0 \end{bmatrix} \cdot I(Z^1 + Z^2 \geq 1),$$

with a sample of size 1,000, Figure 4.2 presents equivalent plots for our tree boosted VCM. These results are less clear since logistic regression produces more volatile gradients.

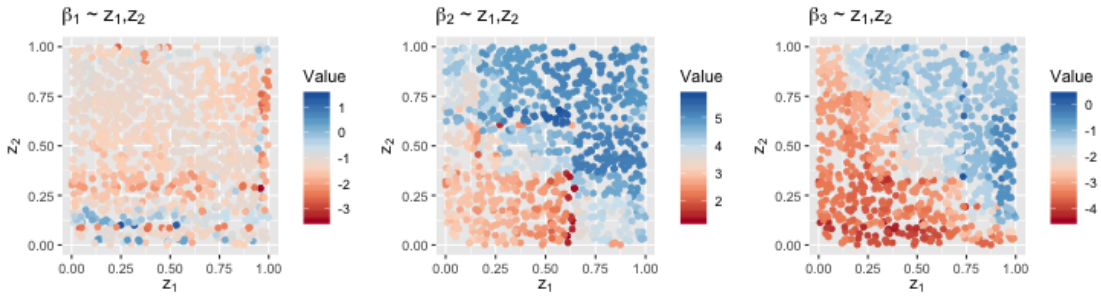


Figure 4.2: Example of varying coefficient mappings on the action space under the logistic regression settings.

In both cases our methods correctly identify $\beta(z)$ as segmenting along the diagonal in z, providing clear visual identification of the behavior of $\beta(z)$. These figures are evidence of the capability of tree boosted VCM to find the varying coefficients without posting structural assumptions on the action space. Further empirical studies are presented in Section 4.

4.3 Asymptotics

There is a large literature providing statistical guarantees and asymptotic analyses of different versions of VCM with varying coefficient mappings obtained via splines or local smoothers (Park et al., 2015; Fan et al., 1999, 2005). In this section we will demonstrate the asymptotic analyses of tree boosted VCM under mild conditions.

4.3.1 Tree Boosted VCM with L^2 Loss

Consider L^2 boosting setting for regression. Given the relationship

$$Y = f(X, Z) + \epsilon_Z, \quad \epsilon_Z \sim N(0, \sigma_Z^2),$$

we work with the following assumptions.

(E1) Unit support of X that supp $X = \{1\} \times [-1, 1]^p$, which is achievable by standardizing without loss of generality for any finitely supported X.

(E2) Uniform bounded noise variance that $\sigma_Z \leq \sigma^*$.

(E3) L^2 loss that $L(u, y) = \frac{1}{2}(u - y)^2$.

Under these conditions, evaluating the pseudo-gradient given in (4.3) yields

$$\Delta_\beta(z_i) = -\nabla_\beta l(y_i, x_i^T\beta)\Big|_{\beta=\hat{\beta}(z_i)} = (y_i - x_i^T\hat{\beta}(z_i)) \cdot x_i.$$

For an existing terminal node $R \subseteq \mathbb{A}$, as per (4.4), the decision tree update in R is

$$\Delta_\beta(z \in R) = \frac{\sum_{i=1}^n (y_i - x_i^T\hat{\beta}(z_i)) \cdot x_i \cdot I(z_i \in R)}{\sum_{i=1}^n I(z_i \in R)}, \tag{4.5}$$

and should subsample w be present

$$\Delta_\beta(z \in R; w) = \frac{\sum_{i=1}^n (y_i - x_i^T\hat{\beta}(z_i)) \cdot x_i \cdot I(z_i \in R)\, I(i \in w)}{\sum_{i=1}^n I(z_i \in R)\, I(i \in w)}.$$

4.3.2 Decomposing Decision Trees

We assume the action space $\mathbb{A}$ involves only continuous and categorical covariates, therefore we will consider its embedding into a Euclidean space $\mathbb{R}^d$ where $d = \dim(\mathbb{A})$ is the dimension of the embedding. Denote $\mathcal{R} = \{(a_1, b_1] \times \cdots \times (a_d, b_d] | -\infty \leq a_i \leq b_i \leq \infty\}$ the collection of all hyper rectangles in $\mathbb{A}$. This set includes all possible terminal nodes of any decision tree built on $\mathbb{A}$. Given the distribution $(Z, 1, X) \sim \mathbb{P}$, we define the inner product $\langle f_1, f_2 \rangle = \mathbb{E}_{\mathbb{P}}[f_1 f_2]$, and the norm $\|\cdot\| = \|\cdot\|_{\mathbb{P},2}$ on the sample space $\mathbb{A} \times \{1\} \times [-1, 1]^p$. For a sample of size n, we write the (unscaled) empirical counterpart by $\langle f_1, f_2 \rangle_n = \sum_{i=1}^n f_1(x_i, z_i) f_2(x_i, z_i)$, such that $n^{-1} \langle f_1, f_2 \rangle_n \to \langle f_1, f_2 \rangle$ by the law of large numbers, with a corresponding norm $\|\cdot\|_n$.

Consider the following classes of functions on $\mathbb{A} \times \{1\} \times [-1, 1]^p$.

- $\mathcal{H} = \{h_R(x, z) = I(z \in R) | R \in \mathcal{R}\}$, indicators of hyper rectangles.
- $\mathcal{G} = \left\{g_{R,j}(x, z) = I(z \in R) \cdot x^j | R \in \mathcal{R}, j = 0, \ldots, p\right\}$ constants and coordinate mappings in hyper rectangles in $\mathcal{R}$. In particular we write $1 = x^0$ so that $g_{R,0} = h_R$.

Bühlmann (2002) established a consistency guarantee for tree-type basis functions for L^2 boosting, in which the key point is to bound the gap between the boosting procedure and its population version by the uniform convergence in distribution of the family of indicators for hyper rectangles. We take a similar approach, for which we have to extend the uniform convergence to a broader function class defined as $\mathcal{G}$ as defined above. The following lemma provides uniform bounds on the asymptotic variability pertaining to $\mathcal{G}$

using Donsker's theorem (see van der Vaart and Wellner, 1996).

Lemma 4.1. *For given L^2 function f and random sub-Gaussian noise ϵ, the following empirical gaps*

1. $\xi_{n,1} = \sup_{R\in\mathcal{R}} \left| \|h_R\|^2 - \frac{1}{n} \|h_R\|_n^2 \right|,$
2. $\xi_{n,2} = \sup_{R\in\mathcal{R}, j=0,\ldots,p} \left| \left\|g_{R,j}\right\|^2 - \frac{1}{n} \left\|g_{R,j}\right\|_n^2 \right|,$
3. $\xi_{n,3} = \sup_{R\in\mathcal{R}, j=0,\ldots,p} \left| \left\langle f, g_{R,j} \right\rangle - \frac{1}{n} \left\langle f, g_{R,j} \right\rangle_n \right|,$
4. $\xi_{n,4} = \sup_{R\in\mathcal{R}, j=0,\ldots,p} \left| \frac{1}{n} \left\langle \epsilon, g_{R,j} \right\rangle_n \right|,$
5. $\xi_{n,5} = \sup_{R_1,R_2\in\mathcal{R}, j,k=0,\ldots,p} \left| \left\langle g_{R_1,j}, g_{R_2,k} \right\rangle - \frac{1}{n} \left\langle g_{R_1,j}, g_{R_2,k} \right\rangle_n \right|,$
6. $\xi_{n,6} = \left| \frac{1}{n} \|f+\epsilon\|_n^2 - \|f+\epsilon\|^2 \right|,$

satisfy that $\xi_n = \max_{i=1}^6 \xi_{n,i} = O_p\left(n^{-\frac{1}{2}}\right)$.

Introduce the empirical remainder function $\hat{r}_b$ such that

$$\hat{r}_0(x,z) = f(x,z) + \epsilon, \quad \hat{r}_b(x,z) = f(x,z) + \epsilon - \hat{\beta}_b(z)^T x, b > 0,$$

i.e. the remainder term after b-th boosting iteration. Further, consider the b-th iteration utilizing $p+1$ decision trees whose disjoint terminal nodes are $R_1^j, \ldots, R_m^j \in \mathcal{R}$ for $j = 0, \ldots, p$ respectively. (4.5) is equivalent to the following expression for the boosting update of the remainder $\hat{r}_b$

$$\hat{r}_{b+1} = \hat{r}_b - \lambda \sum_{i=1}^{m} \sum_{j=0}^{p} \frac{n^{-1} \left\langle \hat{r}_b, g_{R_i^j,j} \right\rangle_n}{n^{-1} \left\| h_{R_i^j} \right\|_n^2} g_{R_i^j,j}, \tag{4.6}$$

or, for simplicity, we flatten the subscripts when there is no ambiguity such that

$$\hat{r}_{b+1} = \hat{r}_b - \lambda \sum_{i=1}^{m(p+1)} \frac{n^{-1} \langle \hat{r}_b, g_{b,i} \rangle_n}{n^{-1} \left\| h_{b,i} \right\|_n^2} g_{b,i},$$

where as defined above, $g_{b,i} = g_{R_i^j,j} = I\left(z \in R_i^j\right) \cdot x^j$. Although the update involves $m(p+1)$ terms, only $p+1$ of them are applicable for a given (x, z) pair as the result of using disjoint terminal nodes.

Further, Mallat and Zhang (1993) and Bühlmann (2002) suggested that we consider the population counterparts of these processes defined by the remainder functions starting with $r_0 = f$ and

$$r_{b+1} = r_b - \lambda \sum_{i=1}^{m(p+1)} \frac{\langle r_b, g_{b,i} \rangle}{\left\| h_{b,i} \right\|^2} g_{b,i}, \tag{4.7}$$

with the same boosted trees used. They concluded that these processes converge to the consistent estimate in the completion of the decision tree family $\mathcal{T}$. As a result, we can achieve asymptotic consistency as long as the gap between the sample process and this population process diminishes fast enough along with the increase of sample size.

4.3.3 Consistency

Lemma 4.1 helps to quantify the discrepancy between tree boosted VCM fits and their population versions conditioned on the sequence of trees used during boosting by decomposing a decision tree having terminal nodes in $\mathcal{R}$ into several hyper rectangles. This strategy also applies to tree boosted VCM. To further achieve consistency, we pose several additional conditions.

(C1) In practice we require the learning rate λ to satisfy that $\lambda \leq (1+p)^{-1}$, while in proofs we use $\lambda = (1+p)^{-1}$.

(C2) All terminal nodes of the trees in the ensemble should have at least N_n observations such that $N_n \geq O\left(n^{\frac{3}{4}+\eta}\right)$ for some small $\eta > 0$, in which case we will have

$$\frac{1}{n}\|h_R\|_n^2 = \frac{1}{n}\sum_{i=1}^{n} I\left(z_i \in R\right) \geq O\left(n^{-\frac{1}{4}+\eta}\right)$$

for all $R \in \mathcal{R}$ that appear as terminal nodes in the ensemble.

(C3) We apply early stopping, allowing at most $B = B(n) = o(\log n)$ iterations during boosting.

(C4) From the optimization perspective, we also require that trees in the ensemble have terminal nodes that effectively reduce the empirical risk. Consider the best functional rectangular fit during the b-th population iteration

$$g^* = \arg\max_{g \in \mathcal{G}} \frac{|\langle r_b, g\rangle|}{\|g\|}.$$

We expect to empirically select at least one (R^*, j) pair during the iteration to approximate g^* such that

$$\frac{|\left\langle r_b, g_{R^*,j}\right\rangle|}{\left\|g_{R^*,j}\right\|} > \nu \cdot \frac{|\langle r_b, g^*\rangle|}{\|g^*\|},$$

for some $0 < \nu < 1$. Lemma 4.1 indicates that by choosing the sample version optimum

$$\hat{g}^* = \arg\max_{g \in \mathcal{G}} \frac{|\langle r_b, g\rangle_n|}{\|g\|_n},$$

the above requirement can be hold true in probability for a fixed number of iterations.

(C5) $\|f\|^2 = M \leq \infty$. In addition, due to the linear models in the VCM, to achieve consistency we require that $f \in \text{span}(\mathcal{G})$.

(C6) We also require the identifiability of linear models such that the distribution of X conditioned on any choice of $Z = z$ should spread uniformly, i.e.

$$\inf_{R\in\mathcal{R},j=0,\ldots,p} \frac{\left\|g_{R,j}\right\|}{\|h_R\|} = \alpha_0 > 0.$$

(C7) A stronger version of (C6) is to assume the existence of $s > 0, c > 0$ s.t. $\forall z$ a.e., there exists an open ball $B_z(x_0, s) \in [-1, 1]^p$ centered at $x_0 = x_0(z)$ inside of which $P(X = (1, x)|Z = z)$ is bounded below by c. In other words, conditioned on any choice of $Z = z$ there is enough spreading sample points in an open region of X that assures model identifiability.

Among all proposed conditions, (C4) is the hardest one to justify using finite sample due to its required optimality. This is when building adaptive trees becomes appealing as to effectively guarantee the optimality in a greedy way with respect to the sample. In contrast, building completely randomized trees is of less an issue asymptotically, as long as the fine segmentation reaches the resolution of detecting micro structures on the action space. This observation refreshes the idea we talked before that the asymptotic analysis of tree methods will favor randomized trees more than adaptive trees.

During local gradient descent, unwanted behaviors can take place when there is local dependent relation between X and Z in the vicinity of some $Z = z$. Extreme cases include $P(X^1 = X^2|Z = z) = 1$, two covariates being collinear, or $P(X^1 = x|Z = z) = 1$, some covariates having degenerate conditional distributions. These cases prevent the local parametric model from being identifiable, and the introduction of (C6) and (C7) avoids those cases.

Theorem 4.2. *Under conditions (C1)-(C5), consider function $f \in span(\mathcal{G})$,*

$$\mathbb{E}_{(x^*,z^*)}\left[|\hat{\beta}_B(z^*)^T x^* - f(x^*,z^*)|^2\right] = o_p(1), n \to \infty,$$

for making predictions at a random point (x^*, z^*) which are independent from but identically distributed as the training data.

Corollary 4.3. *If we further assume (C6),*

$$\hat{\beta}_B(z^*) \xrightarrow{p} \beta(z^*), n \to \infty.$$

Corollary 4.3 justifies the varying coefficient mappings as valid estimators for the true varying linear relationship. Although we have not explicitly introduced any continuity condition on β, it is worth noticing that (C5) requires β to have relatively invariant local behavior. Although one region in $\mathbb{A}$ of any size can be eventually detected by the growing n to fit into a terminal node with sufficient sample points required by (C2), such rate is too loose to guarantee the detection of a small area with a small sample. As a result, tree boosted VCM should be the most ideal when $\mathbb{A}$ is heterogeneous with a few big and flat regions. When we consider the interpretability of tree boosted VCM, consistency is also the sufficient theoretical guarantee for *local fidelity* discussed in Ribeiro et al. (2016) that an interpretable local method should also yield accurate local relation between covariates and responses.

4.4 Empirical Study

4.4.1 Identifying Signals

Our theory suggests that tree boosted VCM is capable of identifying local linear structures and their coefficients accurately. To demonstrate this in practice, we apply it to the following regression problem with higher order feature interaction on the action space.

$$z = (z^1, z^2, z^3, z^4), \quad z^1, z^2 \sim \text{Unif}\{1, \ldots, 10\}, \quad z^3, z^4 \sim \text{Unif}[0, 1].$$

$$x \in \mathbb{R}^7, \quad x \sim N(0, I_7), \quad \epsilon \sim N(0, 0.25).$$

The data generating process is describe by the following pseudo code.

if $z^1 < 4$:	$y = 1 + 3x^1 + 7x^2$
else if $z^1 > 8$:	$y = -5 + 2x^1 + 4x^2 + 6x^3$
else if $z^2 = 1, 3$ or 5 :	$y = 5 + 5x^2 + 5x^3$
else if $z^3 < 0.5$:	$y = 10 + 10x^4$
else if $z^4 < 0.4$:	$y = 10 + 10x^5$
else if $z^3 < z^4$:	$y = 5 - 5x^2 - 10x^3$
else :	$y = -10x^1 + 10x^3$

We utilize a sample of size $10,000$ and use 100 trees of maximal depth of 6 for boosting with constant learning rate of 0.2. Figure 4.3 plots the fitted distribution of each coefficient in red against the ground truth in grey, with reported MSE 3.28. We observe that all

peaks and their intensities properly reflect the coefficient distributions on the action space. Despite the linear expressions, we have tested interaction among all four action covariates of a tree depth of 6 and have not yet achieved convergence, which we conclude as the reasons for large MSE. It manifests the effectiveness of our straightforward implementation of decision trees segmenting the action space.

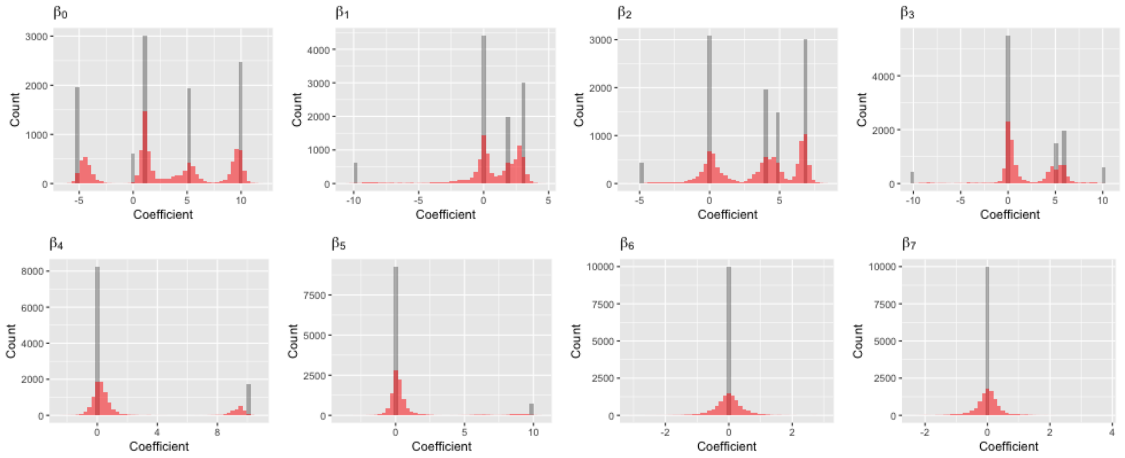


Figure 4.3: Histograms of distributions of fitted coefficient values. Color code: ground truth (grey) and tree boosted VCM (red).

4.4.2 Model Accuracy

To show the accuracy of our proposed methods, we have selected 12 real world datasets and run tree boosted VCM (marked as TVCM) against other benchmark methods. Table 4.1 demonstrates the results under classification settings with three benchmarks: GLM as logistic regression, GLM(S) as a partially saturated logistic regression model where

each combination of discrete action covariates acts as fixed effect with its own level, and AdaBoost. Table 4.2 demonstrates the results under regression settings. The three benchmarks we choose here are: LM as linear model, LM(S) as a partially saturated linear model, and GBM as the gradient boosted trees. Although with additional structural assumptions, tree boosted VCM performs nearly on a par with both GBM and AdaBoost. It benefits from its capability of modeling the action space without structural conditions to outperform the fixed effect linear model in certain cases.

NAME	GLM	GLM(S)	ADABOOST	TVCM
MAGIC04	0.208(0.007)	0.209(0.0065)	0.13(0.0076)	0.209(0.0072)
BANK	0.111(0.0044)	0.1(0.0044)	0.098(0.0035)	0.114(0.0043)
OCCUPANCY	0.014(0.0042)	0.0129(0.0034)	0.00567(0.0016)	0.0126(0.0048)
SPAMBASE	0.0749(0.011)	0.0732(0.012)	0.0564(0.0098)	0.0616(0.0097)
ADULT	0.188(0.004)	0.155(0.0046)	0.136(0.0049)	0.154(0.0028)
EGRIDSTAB	0.289(0.014)	0.227(0.018)	0.179(0.009)	0.177(0.015)

Table 4.1: Prediction accuracy of classification and 0-1 loss for six UCI data sets through tenfold cross validation. Results are shown as mean(sd). Sources of some datasets are: BANK(Moro et al., 2014) and OCCUPANCY(Candanedo and Feldheim, 2016).

4.4.3 Visual Interpretability: Beijing Housing Price

Here we show the results of applying tree boosted VCM on the Beijing housing data (Kaggle, 2018). We take the housing unit price as the target regressed on covariates of location, floor, number of living rooms and bathrooms, whether the unit has an elevator and whether the unit has been refurbished. Specially, location has been treated as the action space represented in pairs of longitude and latitude. Location specific linear coefficients

NAME	LM	LM(S)	GBM	TVCM
BEIJINGPM	6478(227)	5041(203)	3465(176)	3942(178)
BIKEHOUR	24590(1630)	12190(818)	5791(419)	6596(597)
STARCRAFT	1.135(0.0622)	1.116(0.0645)	1.045(0.0594)	1.161(0.0596)
ONLINENEWS	0.8544(0.0331)	0.8377(0.0328)	0.7826(0.0298)	0.8183(0.0337)
ENERGY	18.01(4.42)	9.801(2.16)	0.5633(0.162)	9.864(2.27)
EGRIDSTAB	1.01e-03(4.5e-05)	6.92e-04(3e-05)	4.31e-04(1.4e-05)	4.27e-04(8.3e-06)

Table 4.2: Prediction accuracy of regression and mean square error for six UCI data sets through tenfold cross validation. Results are shown as mean(sd). Sources of some datasets are: BEIJINGPM(Liang et al., 2015), BIKEHOUR(Fanaee-T and Gama, 2014), ONLINENEWS(Fernandes et al., 2015) and ENERGY(Tsanas and Xifara, 2012).

of other covariates are displayed in Figure 4.4. We allow 200 trees of depth of 5 in the ensemble with a constant learning rate of 0.05.

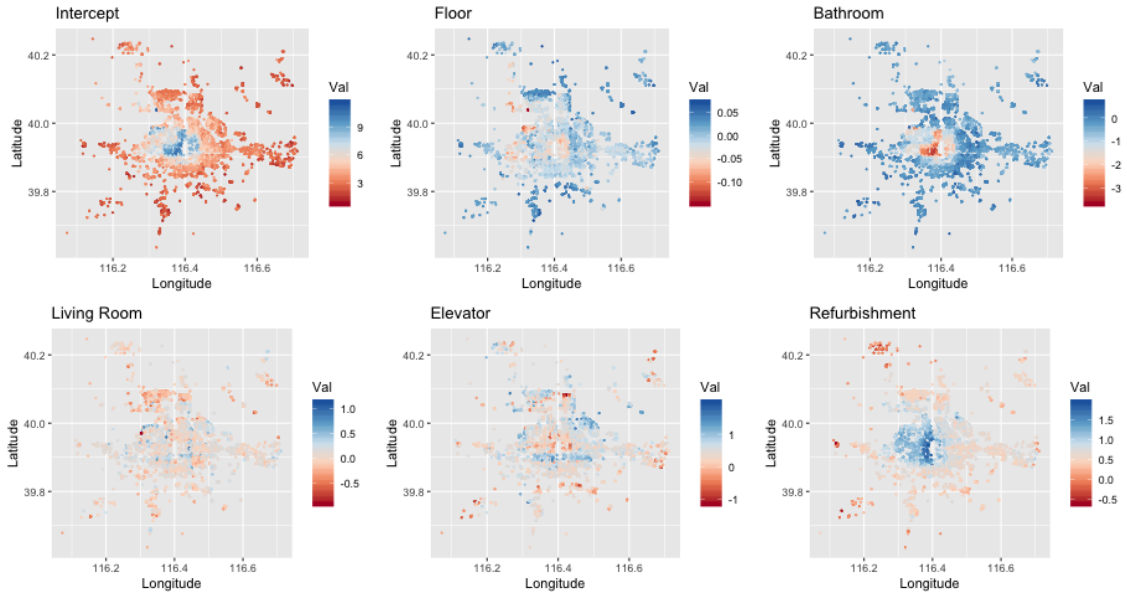


Figure 4.4: Beijing housing unit price broken down on several factors.

The urban landscape of Beijing is pictured by its old inner circle with a low skyline

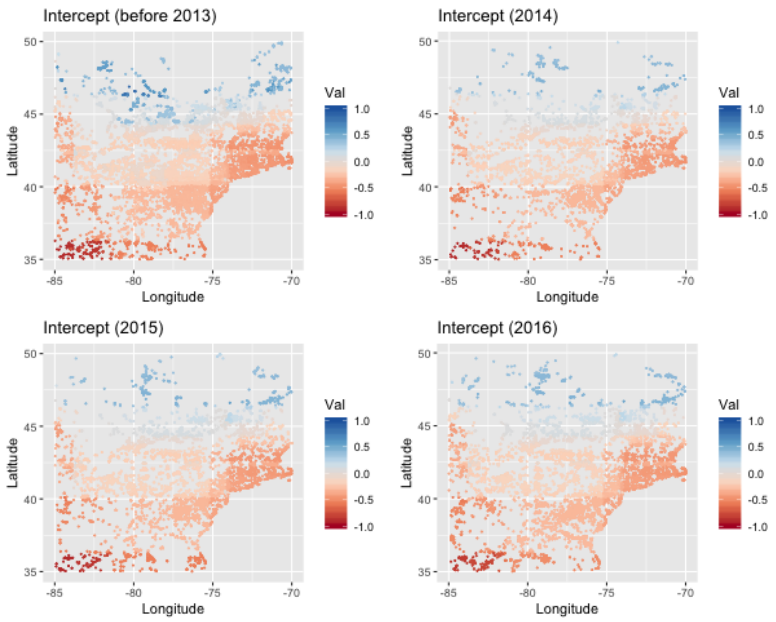
gradually transitioning to its modern outskirt rim of skyscrapers housing a young and new workforce. Our model intercept provides the baseline of the unit housing prices in each area. Despite their high values, most buildings inside the inner circle are old and not suitable for replanning, so elevators and number of bathrooms are of low contribution to the final price, while their refurbishment gets more attention. In contrast, outskirt housing gains more value if the unit has a complementary elevator and is on higher floor.

Figure 4.4 provides clear visualization of the fitted tree boosted VCM. Usually these irregular patterns are more likely to be outputs of nonparametric models, while behind each point on our plot is a location-specific linear model predicting the housing price breaking down to different factors.

4.4.4 Fitting Other Model Class

As mentioned, since VCM is the generalization of many specific models, our proposed fitting algorithm and analysis should apply to them as well. We take partially linear models as an example and consider the following data set from Cornell Lab of Ornithology consisting of the recorded observations of four species of vireos along with the location and surrounding terrain types. We apply a tree boosted VCM under logistic regression setting using longitude, latitude and year as the action space and all rest covariates as linear effects, obtaining the model demonstrated by Table 4.4.4. The intercept plot suggests the trend of observed vireos favoring cold climate and inland environment, while the slopes of different territory types indicate a strong preference towards the low elevation between de-

ciduous forests and evergreen needles. It can also be used to compare the baselines across different years in the past decade.



Covariate	Slope
Elevation	9.65e-04
Shallow Ocean	-1.88e+03
CoastShore Lines	-6.51e+01
Shallow Inland	9.39e+01
Moderate Ocean	-1.18e+03
Deep Ocean	-5.12e+03
Evergreen Needle	-4.54e+02
Grasslands	-4.49e+02
Croplands	-4.29e+02
Urban Built	-6.62e+02
Barren	-1.59e+03
Evergreen Broad	2.77e+02
Deciduous Needle	2.57e+02
Deciduous Broad	2.72e+02
Mixed Forest	7.32e+01
Closed Shrubland	-1.19e+03
Open Shrubland	8.60e+01
Woody Savannas	-5.75e+02
Savannas	-7.46e+02

Table 4.3: Fitting a partially linear model using tree boosted VCM. Plot on the left shows the nonparametric intercept. Table on the right shows the coefficients of predictive covariates.

4.5 Shrinkage, Selection and Serialization

Tree boosted VCM is compatible with any alternative boosting strategy in place of the boosting steps (B3) and (B4), such as the use of subsampled trees (Friedman, 2002), univariate or bivariate trees (Lou et al., 2012; Hothorn et al., 2013) or adaptive shrinkage (dropout) (Rashmi and Gilad-Bachrach, 2015; Rogozhnikov and Likhomanenko, 2017).

While these alternative approaches have been empirically shown to help avoid overfitting or provide more model interpretability, we also anticipate that the corresponding varying coefficient mappings would inherit certain theoretical properties. For instance, our Boulevard boosting guarantees finite sample convergence and asymptotic normality of its predictions. Incorporating Boulevard into our tree boosted VCM framework requires the changes to (B3) and (B4) such that

(B3*) For each j, find a good fit $t^j_{b+1} \in \mathcal{T} : \mathbb{A} \to \mathbb{R}$ on $(z_i, \Delta^j_{\beta_i})$ for $i \in w \subset \{1, \ldots, n\}$ a random subsample.

(B4*) Update $\hat{\beta}_b$ with learning rate $\lambda < 1$.

$$\hat{\beta}_{b+1}(\cdot) = \frac{b}{b+1}\hat{\beta}_b(\cdot) + \frac{\lambda}{b+1}\Gamma_M\left(\begin{bmatrix} t^0_{b+1}(\cdot) \\ \vdots \\ t^p_{b+1}(\cdot) \end{bmatrix}\right),$$

where Γ_M truncates the absolute value at some $M > 0$.

By taking the same approach in the original paper, we can show that boosting VCM with Boulevard will also yield finite sample convergence to a fixed point.

Boulevard modifies the standard boosting strategy to the extent that new theoretical results have to be developed specifically. In contrast, there are other boosting variations that fall directly under the theoretical umbrella of tree boosted VCM. Our discussion so far assumes we run boosting iterations with a distinct tree built for each coefficient component while these trees are simultaneously constructed using the same batch of pseudo-residuals. Despite the possibility to utilize a single decision tree with multidimensional response to

produce all components, as long as we build separate trees sequentially, the question arises that whether we should update the pseudo-residuals on the fly.

One advantage of doing so is the minimized boosting iteration from $(1 + p)$ trees down to one tree, allowing us to use much larger learning rate $\lambda \leq 1/2$ instead of $\lambda \leq (1 + p)^{-1}$ without changing the arguments we used to establish the consistency. We also anticipate that doing so in practice moderately reduces the cost as the gradients become more accurate for each tree. Here we will consider two approaches to conduct the on-the-fly updates.

In Hothorn et al. (2013) the authors proposed the component-wise linear least squares for boosting where they select which β to update using the stepwise optimal strategy, i.e., choose j_b and update β^{j_b} if

$$j_b = \arg\min_{j=0,\ldots,p} \sum_{i=1}^{n} l(y_i, x_i^T(\hat{\beta}_b + \lambda t_b^j e_j)(z_i)),$$

the component tree that reduces the empirical risk the most. As a result, (B4) in Algorithm now updates

$$\hat{\beta}_{b+1} = \hat{\beta}_b + \lambda t_{b+1}^{j_b} e_{j_b}.$$

Notice that finding this optimum still requires the comparison among all components, therefore does not save any training cost when there are no better means or prior knowledge to help detect which component stands out. That being said, the optimal move is compatible with the key condition (C4) we posed to ensure consistency. Namely, it still guarantees that the population counterpart of boosting is efficient in reducing the gap between the estimate and the truth. However, this greedy strategy also complicates the pattern of the sequence in which β's get updated.

Serialization refers to the cases when the β's are being updated in some predetermined order. A similar model is covered by Lou et al. (2012, 2013) where the authors applied univariate generalized additive models (GAM) to perform model distillation, which was refined in Tan et al. (2017) using decision trees. Their models can either be built through backfitting which eventually produces one additive component for each covariate, or through boosting that generates a sequence of additive terms.

Applying the rotation of coordinates to tree boosted VCM, we can break each of the original boosting iterations into (p+1) micro steps to write

$$\hat{\beta}_{b,j} = \hat{\beta}_{b,j-1} - \lambda \nabla_{\beta^j} l(y, x^T \beta)\Big|_{\beta=\hat{\beta}_{b,j-1}(z)},$$

with j rotating through $0, \ldots, p$. This procedure immediately updates the pseudo-residuals after each component tree is built. There are two feasible approaches if we intend to employ tree boosted VCM to achieve the same univariate GAM model. Either we can place all covariates into the action space and use only univariate decision trees to perform the serialized boosting, or we can directly apply tree boosted VCM to get additive models that are univariate with respect to the predictive covariates.

However, this procedure is not compatible with our consistency conclusion as the serialized boosting fails to guarantee (C4): each micro boosting step on a single coordinate relies on the current pseudo gradients instead of the gradients before the entire rotation. One solution is to consider an alternative to the determined updating sequence by randomly and uniformly proposing the coordinate to boost. In this regard,

$$\hat{\beta}_b = \hat{\beta}_b - \lambda \nabla_{\beta^j} l(y, x^T \beta)\Big|_{\beta=\hat{\beta}_b(z)},$$

where $j \sim \text{Unif}\{0, \ldots, p\}$. This stochastic sequence solves the compatibility issue by satisfying (C4) with a probability bounded from below.

4.6 Proofs

In this section we list the complete proofs of all theorems covered above.

Proof to Lemma 4.1

Proof. $\xi_{n,6}$ is simply CLT. For the rest, we will conclude the corresponding function classes are $P-$Donsker. The collection of indicators for hyper rectangles $(-\infty, a_1]\times\ldots,(-\infty, a_p] \subseteq \mathbb{R}^p$ is Donsker. By taking difference at most p times we get all elements in $\mathcal{H}$, therefore $\mathcal{G}$, the indicators of $\mathcal{R}$, is Donsker. Thus $\xi_{n,1} = O_p\left(n^{-\frac{1}{2}}\right)$.

The basis functions $\mathcal{E} = \{1, x^j, j = 1, \ldots, p\}$ is Donsker since all elements are monotonic and bounded since $x \in [-1, 1]^p$. So $\mathcal{G} = \mathcal{H} \times \mathcal{E}$ is Donsker, which gives $\xi_{n,2} = O_p\left(n^{-\frac{1}{2}}\right)$ and $\xi_{n,4} = O_p\left(n^{-\frac{1}{2}}\right)$.

In addition, for fixed f, $f\mathcal{G}$ is therefore Donsker, which gives $\xi_{n,3} = O_p\left(n^{-\frac{1}{2}}\right)$. And $\mathcal{G} \times \mathcal{G}$ is Donsker, which gives $\xi_{n,5} = O_p\left(n^{-\frac{1}{2}}\right)$.

□

Proof to Theorem 4.2

To supplement our discussion of norms, it is immediate that $\left\|g_{R,j}\right\| \leq \|h_R\| \leq 1$. Another key relation is $\left\|g_{R,j}\right\|_{n,1} \leq \|h_R\|_{n,1} = \|h_R\|_n^2$. We also assume that all R's satisfy the terminal node condition.

Lemma 4.4. $\|\hat{r}_{b+1}\|_n \leq \|\hat{r}_b\|_n$, $\|r_{b+1}\| \leq \|r_b\|$.

Proof. Consider the $p+1$ trees used for one boosting iteration with the terminal nodes denoted as $R_i^j, 0 = 1, \ldots, p, i = 1, \ldots, m$, write $I_R = I(z \in R)$.

$$
\begin{aligned}
\|\hat{r}_{b+1}\|_n &= \left\| \hat{r}_b - \lambda \sum_{j=0}^{p} \sum_{i=1}^{m} \frac{n^{-1} \left\langle \hat{r}_b, g_{R_i^j,j} \right\rangle_n}{n^{-1} \left\| h_{R_i^j} \right\|_n^2} g_{R_i^j,j} \right\|_n \\
&\leq \sum_{j=0}^{p} \left\| \sum_{i=1}^{m} \left(\lambda \hat{r}_b I_{R_i^j} - \frac{n^{-1} \left\langle \lambda \hat{r}_b I_{R_i^j}, g_{R_i^j,j} \right\rangle_n}{n^{-1} \left\| h_{R_i^j} \right\|_n^2} g_{R_i^j,j} \right) \right\|_n \\
&= \sum_{j=0}^{p} \left(\left\| \sum_{i=1}^{m} \left(\lambda \hat{r}_b I_{R_i^j} - \frac{n^{-1} \left\langle \lambda \hat{r}_b I_{R_i^j}, g_{R_i^j,j} \right\rangle_n}{n^{-1} \left\| h_{R_i^j} \right\|_n^2} g_{R_i^j,j} \right) \right\|_n^2 \right)^{\frac{1}{2}} \\
&= \sum_{j=0}^{p} \left(\sum_{i=1}^{m} \left\| \lambda \hat{r}_b I_{R_i^j} - \frac{n^{-1} \left\langle \lambda \hat{r}_b I_{R_i^j}, g_{R_i^j,j} \right\rangle_n}{n^{-1} \left\| h_{R_i^j} \right\|_n^2} g_{R_i^j,j} \right\|_n^2 \right)^{\frac{1}{2}} \\
&\leq \sum_{j=0}^{p} \left(\sum_{i=1}^{m} \left\| \lambda \hat{r}_b I_{R_i^j} \right\|_n^2 \right)^{\frac{1}{2}} = \sum_{j=0}^{p} \|\lambda \hat{r}_b\|_n = \|\hat{r}_b\|_n ,
\end{aligned}
$$

given $\left\| h_{R_i^j} \right\|_n^2 \geq \left\| g_{R_i^j,j} \right\|_n^2$. Same argument can be applied to the population version hence we get the second part.

□

Lemma 4.5. *For any $b \leq 0$, as defined in (4.7),*

$$\sup_{x,z} |r_{b+1}(x,z)| \leq 2 \sup_{x,z} |r_b(x,z)|.$$

Proof. As implied by (4.6), for (x,z) such that $z \in R$,

$$r_{b+1}(x,z) = r_b(x,z) - \lambda \sum_{i=0}^{p} \frac{\left\langle r_b, g_{R,j} \right\rangle}{\|h_R\|^2} g_{R,j}(x,z).$$

The key observation is that

$$\left| \frac{\langle r_b, g_{R,i} \rangle}{\|h_R\|^2} \right| = \left| \frac{\int r_b I\,(z \in R)\, x^j dP}{\int I\,(z \in R)^2\, dP} \right| \leq \sup_{x,z} |r_b(x,z)|.$$

Therefore, provided $|g_{R,i}| \leq 1$ and write $z \in R_z$,

$$\begin{aligned}
\sup_{x,z} |r_{b+1}| &\leq \sup_{x,z} |r_b| + \lambda \sup_{x,z} \sum_{i=0}^{p} \left| \frac{\left\langle r_b, g_{R_z,i} \right\rangle}{\left\| h_{R_z} \right\|^2} \right| \left| g_{R_z,i} \right| \\
&\leq \sup |r_b| + \lambda \sum_{i=0}^{p} \sup |r_b| \cdot 1 \\
&= 2 \sup_{x,z} |r_b|.
\end{aligned}$$

Recursively we can conclude that $\sup_{x,z} |r_b| \leq 2^b \sup_{x,z} |r_0|$. □

Lemma 4.6. *Under conditions (C1)-(C6),*

$$\|\hat{r}_B\|^2 = \|r_B\|^2 + \sigma_\epsilon^2 + o_p(1),$$

where $\sigma_\epsilon^2 = \|\epsilon\|^2$.

Proof. Recall that

$$\hat{r}_{b+1} = \hat{r}_b - \lambda \sum_{j=0}^{p} \sum_{i=1}^{m} \frac{n^{-1} \left\langle \hat{r}_b, g_{R_i^j,j} \right\rangle_n}{n^{-1} \left\| h_{R_i^j} \right\|_n^2} g_{R_i^j,j} = \hat{r}_b - \lambda \sum_{i=1}^{m(p+1)} \frac{n^{-1} \left\langle \hat{r}_b, g_{b,i} \right\rangle_n}{n^{-1} \left\| h_{b,i} \right\|_n^2} g_{b,i},$$

and

$$r_{b+1} = r_b - \lambda \sum_{j=0}^{p} \sum_{i=1}^{m} \frac{\left\langle r_b, g_{R_i^j,j} \right\rangle}{\left\| h_{R_i^j} \right\|^2} g_{R_i^j,j} = r_b - \lambda \sum_{i=1}^{m(p+1)} \frac{\left\langle r_b, g_{b,i} \right\rangle}{\left\| h_{b,i} \right\|^2} g_{b,i}.$$

Therefore

$$\begin{aligned}
\hat{r}_{b+1} - r_{b+1} &= (\hat{r}_b - r_b) + \lambda \sum_{j=0}^{p} \sum_{i=1}^{m} \left(\frac{\left\langle r_b, g_{R_i^j,j} \right\rangle}{\left\| h_{R_i^j} \right\|^2} - \frac{n^{-1} \left\langle \hat{r}_b, g_{R_i^j,j} \right\rangle_n}{n^{-1} \left\| h_{R_i^j} \right\|_n^2} \right) g_{R_i^j,j} \\
&= (\hat{r}_b - r_b) + \lambda \sum_{i=1}^{m(p+1)} \left(\frac{\left\langle r_b, g_{b,i} \right\rangle}{\left\| h_{b,i} \right\|^2} - \frac{n^{-1} \left\langle \hat{r}_b, g_{b,i} \right\rangle_n}{n^{-1} \left\| h_{b,i} \right\|_n^2} \right) g_{b,i} \\
&\triangleq (\hat{r}_b - r_b) + \lambda \delta_b \\
&= (\hat{r}_0 - r_0) + \lambda \sum_{j=0}^{b} \delta_j = \epsilon + \lambda \sum_{j=0}^{b} \delta_j.
\end{aligned}$$

Since for each fixed j, all R_i^j are disjoint, we therefore define that

$$\gamma_b = \sum_{j=0}^{p} \sup_{i=1,\ldots,m} \left| \frac{\left\langle r_b, g_{R_i^j,j} \right\rangle}{\left\| h_{R_i^j} \right\|^2} - \frac{n^{-1} \left\langle \hat{r}_b, g_{R_i^j,j} \right\rangle_n}{n^{-1} \left\| h_{R_i^j} \right\|_n^2} \right|,$$

which guarantees $\sup_{x,z} |\delta_b| \leq \gamma_b$. To bound γ_b, without loss of generality, we consider a single term involved such that

$$\begin{aligned}
\frac{\langle r_b, g_b \rangle}{\|h_b\|^2} - \frac{n^{-1} \langle \hat{r}_b, g_b \rangle_n}{n^{-1} \|h_b\|_n^2} &\triangleq \left(\frac{u}{v} - \frac{\hat{u}}{\hat{v}} \right) \\
&= \left(\frac{u - \hat{u}}{v} + \left(\frac{1}{v} - \frac{1}{\hat{v}} \right) \hat{u} \right).
\end{aligned}$$

First consider

$$\begin{aligned}\hat{u} - u &= \frac{1}{n}\langle \hat{r}_b, g_b\rangle_n - \langle r_b, g_b\rangle \\ &= \frac{1}{n}\left\langle \epsilon + r_b + \sum_{j=0}^{b-1}\delta_j, g_b\right\rangle_n - \langle r_b, g_b\rangle \\ &= \frac{1}{n}\langle \epsilon, g_b\rangle_n + \left(\frac{1}{n}\langle r_b, g_b\rangle_n - \langle r_b, g_b\rangle\right) + \left(\sum_{j=0}^{b-1}\frac{1}{n}\left\langle \delta_j, g_b\right\rangle_n\right).\end{aligned}$$

Per Lemma 4.5, we have

$$\left|\frac{1}{n}\langle \epsilon, g_b\rangle_n\right| \le \xi_n$$

and, by iteratively applying Lemma 4.5 and setting $C_0 = \max(\sup_{x,z}|f|, 1)$,

$$\begin{aligned}&\left|\frac{1}{n}\langle r_b, g_b\rangle_n - \langle r_b, g_b\rangle\right| \\ =&\left|\frac{1}{n}\left\langle f - \lambda\sum_{j=0}^{b-1}\sum_{i=1}^{m(p+1)}\frac{\left\langle r_j, g_{j,i}\right\rangle}{\left\|h_{j,i}\right\|^2}g_{j,i}, g_b\right\rangle_n - \left\langle f - \lambda\sum_{j=0}^{b-1}\sum_{i=1}^{m(p+1)}\frac{\left\langle r_j, g_{j,i}\right\rangle}{\left\|h_{j,i}\right\|^2}g_{j,i}, g_b\right\rangle\right| \\ \le&\left|\frac{1}{n}\langle f, g_b\rangle_n - \langle f, g_b\rangle\right| + \lambda\sum_{j=0}^{b-1}\sum_{i=1}^{m(p+1)}\left|\frac{1}{n}\left\langle\frac{\left\langle r_j, g_{j,i}\right\rangle}{\left\|h_{j,i}\right\|^2}g_{j,i}, g_b\right\rangle_n - \left\langle\frac{\left\langle r_j, g_{j,i}\right\rangle}{\left\|h_{j,i}\right\|^2}g_{j,i}, g_b\right\rangle\right| \\ \le&\left|\frac{1}{n}\langle f, g_b\rangle_n - \langle f, g_b\rangle\right| + \lambda\sum_{j=0}^{b-1}\sum_{i=1}^{m(p+1)}\left|\frac{\left\langle r_j, g_{j,i}\right\rangle}{\left\|h_{j,i}\right\|^2}\right|\left|\frac{1}{n}\left\langle g_{j,i}, g_b\right\rangle_n - \left\langle g_{j,i}, g_b\right\rangle\right| \\ \le&\xi_n + \lambda\sum_{j=0}^{b-1}\sup|r_j|m(p+1)\xi_n \\ \le&\xi_n + C_0\sum_{j=0}^{b-1}2^j m\xi_n \\ \le&C_0 2^b m\xi_n.\end{aligned}$$

The last term could be bounded by

$$\left|\sum_{j=0}^{b-1}\frac{1}{n}\left\langle \delta_j, g_b\right\rangle_n\right| \le \frac{1}{n}\sum_{j=0}^{b-1}\left\|\delta_j\right\|_{n,\infty}\|g_b\|_{n,1} \le \frac{1}{n}\sum_{j=0}^{b-1}\gamma_j\|h_b\|_n^2,$$

where

$$\|g_b\|_{n,1} = \sum_{i=1}^{n} |g_b(x_i)|, \quad \left\|\delta_j\right\|_{n,\infty} = \sup_{i=1,\ldots,n} |\delta_j(x_i)|.$$

Hence

$$|\hat{u} - u| \leq C_0 2^b m \xi_n + \frac{1}{n} \sum_{j=0}^{b-1} \gamma_j \|h_b\|_n^2 .$$

In order to bound $|\hat{u}|$, we notice

$$\begin{aligned}
|\hat{u}| = \left| \frac{1}{n} \langle \hat{r}_b, g_b \rangle_n \right| &\leq \left(\frac{1}{n} \|\hat{r}_n\|_n^2 \right)^{\frac{1}{2}} \cdot \left(\frac{1}{n} \|g_b\|_n^2 \right)^{\frac{1}{2}} \\
&\leq \left(\frac{1}{n} \|\hat{r}_0\|_n^2 \right)^{\frac{1}{2}} \cdot \left(\frac{1}{n} \|g_b\|_n^2 \right)^{\frac{1}{2}} \\
&= \left(\frac{1}{n} \|f + \epsilon\|_n^2 \right)^{\frac{1}{2}} \cdot \left(\frac{1}{n} \|g_b\|_n^2 \right)^{\frac{1}{2}} \\
&\leq (M + \sigma_\epsilon^2 + \xi_n) \cdot \|g_b\| \\
&\leq (M_0 + \xi_n) \cdot \|h_b\| .
\end{aligned}$$

Therefore, we get an upper bound for

$$\begin{aligned}
\left| \frac{\langle r_b, g_b \rangle}{\|h_b\|^2} - \frac{n^{-1} \langle \hat{r}_b, g_b \rangle_n}{n^{-1} \|h_b\|_n^2} \right| &\leq \frac{|\hat{u} - u|}{|v|} + \left| \frac{1}{v} - \frac{1}{\hat{v}} \right| |\hat{u}| \\
&= \frac{C_0 2^b m \xi_n + n^{-1} \sum_{j=0}^{b-1} \gamma_j \|h_b\|_n^2}{\|h_b\|^2} + \frac{\xi_n \cdot (M_0 + \xi_n) \cdot \|h_b\|}{\|h_b\|^2 \cdot n^{-1} \|h_b\|_n^2} \\
&\leq \frac{C_0 2^b m \xi_n}{\|h_b\|^2} + \sum_{j=0}^{b-1} \gamma_j \left(1 + \frac{\xi_n}{\|h_b\|^2} \right) + \frac{\xi_n (M_0 + \xi_n)}{\|h_b\| \left(\|h_b\|^2 - \xi_n \right)}.
\end{aligned}$$

Denote h be the global minimum of the ensemble that $h = \min_{b,i,j} \left\| h_{R_i^j} \right\|$, since $m \leq (h^2 - \xi_n)^{-1}$, we obtain

$$\gamma_b \leq (p+1) \left(\frac{C_0 2^b \xi_n}{h^2 (h^2 - \xi_n)} + \sum_{j=0}^{b-1} \gamma_j \left(1 + \frac{\xi_n}{h^2} \right) + \frac{\xi_n (M_0 + \xi_n)}{h (h^2 - \xi_n)} \right).$$

We would like to mention the elementary result that for a series $\{x_n\}$ satisfying

$$x_n \leq 2^n a + \sum_{i=0}^{n-1} b x_i + c,$$

the partial sums satisfy

$$\sum_{i=0}^{n} x_n \leq a \left(\frac{1 - \left(\frac{2}{1+b}\right)^{n+1}}{1 - \frac{2}{1+b}} \right) (1+b)^n - \frac{c}{b}.$$

Hence, we can verify this upper bound that

$$\sum_{j=0}^{B-1} \gamma_j \leq (1+p)^B \left(\frac{C_0}{h^2 - \xi_n} \left(2 + \frac{\xi_n}{h^2}\right)^B \left(1 - \left(1 - \frac{\frac{\xi_n}{h^2}}{2 + \frac{\xi_n}{h^2}} \right)^{B-1} \right) - \frac{\xi_n (M_0 + \xi_n)}{h(h^2 - \xi_n)\left(1 + \frac{\xi_n}{h^2}\right)} \right)$$

Recall the rates that $B = o(\log n), h^2 = O_p\left(n^{-\frac{1}{4}+\eta}\right), \xi_n = O_p\left(n^{-\frac{1}{2}}\right)$, thus

$$\left(2 + \frac{\xi_n}{h^2}\right)^B = 2^B \cdot O_p(1), \quad 1 - \left(1 - \frac{\frac{\xi_n}{h^2}}{2 + \frac{\xi_n}{h^2}} \right)^{B-1} = \frac{\xi_n}{h^2} \cdot O_p(1),$$
$$\frac{\xi_n (M_0 + \xi_n)}{h(h^2 - \xi_n)\left(1 + \frac{\xi_n}{h^2}\right)} = \frac{\xi_n}{h^3} \cdot O_p(1).$$

Hence,

$$\sum_{j=0}^{B-1} \gamma_j \leq (1+p)^B \left(\frac{C_0}{h^2} \cdot 2^B \cdot \frac{\xi_n}{h^2} - \frac{\xi_n}{h^3} \right) O_p(1) = o_p(1),$$

which is equivalent to

$$\left\| \sum_{j=0}^{B-1} \delta_j \right\| \leq \sum_{j=0}^{B-1} \left\| \delta_j \right\| \leq \sum_{j=0}^{B-1} \gamma_j = o_p(1).$$

Combining all above we have

$$\|\hat{r}_B\|^2 = \left\| r_B + \epsilon + \lambda \sum_{j=0}^{B-1} \delta j \right\|^2$$

$$\leq \|\epsilon\|^2 + \|r_B\|^2 + \lambda^2 \left\| \sum_{j=0}^{B-1} \delta_j \right\|^2 + 2\lambda \|r_B + \epsilon\| \left\| \sum_{j=0}^{B-1} \delta_j \right\|$$
$$= \sigma_\epsilon^2 + \|r_B\|^2 + o_p(1).$$

□

Lemma 4.7. *Under condition (C1)-(C6), for any $\rho > 0$ there exists $B_0 = B_0(\rho)$ and $n_0 = n_0(\rho)$ such that for all $n > n_0$,*

$$P\left(\left\|r_{B_0}\right\| \leq \rho\right) \geq 1 - \rho.$$

Proof. Lemma 3 in Bühlmann (2002) proves this statement for rectangular indicators. By fixing $\lambda = (1 + p)^{-1}$ and introducing conditions (C3) and (C4), formula (11) in Bühlmann (2002) still holds in terms of the single terminal node in each of the trees that corresponds to our defined R^*. Therefore cited Lemma 3 holds for our boosted trees. The conclusion is therefore reached by the assumption that $f \in \text{span}(\mathcal{G})$. □

Proof to main Theorem. For a given $\rho > 0$, since $\hat{r}_B(x^*, z^*) - f(x^*, z^*)$ is independent of ϵ,

$$\begin{aligned}
\mathbb{E}_{(x^*,z^*)}\left[|\hat{\beta}_B(z^*)^T x^* - f(x^*, z^*)|^2\right] &= \mathbb{E}_{(x^*,z^*)}\left[|\hat{r}_B(x^*, z^*) - f(x^*, z^*) - \epsilon|^2\right] - \|\epsilon\|^2 \\
&= \|\hat{r}_B\|^2 - \|\epsilon\|^2 \\
&\leq \|r_B\|^2 + o_p(1) \\
&\leq \left\|r_{B_0}\right\|^2 + o_p(1) \\
&\leq \rho O_p(1) + o_p(1).
\end{aligned}$$

We reach the conclusion by sending $\rho \to 0$. □

Proof to Corollary 4.3

Proof. We prove by contradiction. Assume there exists $0 < \epsilon_0 < s, c_0 > 0$ s.t.

$$P(\|\beta_B(z^*) - \beta(z^*)\|^2 > \epsilon_0) \geq c_0$$

for any sufficiently large n. Fix n and consider any z_0 s.t. $\|\beta_B(z_0) - \beta(z_0)\| > \epsilon_0$. The corresponding open ball $B(x_0, s)$ has volume $v_0 = O(s^p)$. Write $\beta = \begin{bmatrix} \beta^0 \\ \beta^{-0} \end{bmatrix}$,

$$\begin{aligned}
&\int_{B(x_0,s)} \left\langle \begin{bmatrix} 1 \\ x \end{bmatrix}, \beta_B(z_0) - \beta(z_0) \right\rangle^2 dP_{x|z_0} \\
\geq & c \int_{B(x_0,s)} \left\langle \begin{bmatrix} 1 \\ x \end{bmatrix}, \beta_B(z_0) - \beta(z_0) \right\rangle^2 dx \\
\geq & c v_0 \left\langle \begin{bmatrix} 1 \\ x_0 \end{bmatrix}, \beta_B(z_0) - \beta(z_0) \right\rangle^2 + c \int_{B(0,s)} \left\langle \begin{bmatrix} 1 \\ x \end{bmatrix}, \beta_B(z_0) - \beta(z_0) \right\rangle^2 dx \\
\geq & c v_0 (\beta_B(z_0)^0 - \beta(z_0)^0)^2 + c t_0 \left\| \beta_B(z_0)^{-0} - \beta(z_0)^{-0} \right\|^2 \\
\geq & c \min(v_0, t_0) \epsilon_0.
\end{aligned}$$

where $t_0 = \int_{B(0,s)} x_1^2 dx = O(s^p)$. That is equivalent to

$$\mathbb{E}_{(x^*,z^*)} \left[|\hat{\beta}_B(z^*)^T x^* - f(x^*, z^*)|^2 \right] > c \min(v_0, t_0) \epsilon_0,$$

contradicting Theorem 4.2. □

CHAPTER 5

DISCUSSION AND POTENTIAL FUTURE WORK

5.1 U-statistics and Boosting

The design of Boulevard boosting reflects our intention to create a tree ensemble whose component trees are equally weighted. This idea originates from the successful analysis of random forests using U-statistics. A U-statistic, defined as the average of the exhaustive permutation of a symmetric estimate kernel function h_k, takes the following form as an estimator for an unknown coefficient which in our case is the prediction at a new point,

$$U(x_1, \ldots, x_n) = \frac{1}{\binom{n}{k}} \sum_{x'_1, \ldots, x'_k} h_k(x'_1, \ldots, x'_k),$$

where $x'_1, \ldots, x'_k$ iterates through all combinations of k elements in $x_1, \ldots, x_n$.

While U-statistics directly yields asymptotic normality, its incomplete version which averages a subset of all possible permutations, and infinite order version whose kernel size inflates with the sample size at certain rate, produce similar results (Van der Vaart, 2000; Mentch and Hooker, 2014). These generalizations can be applied to random forests after adjusting for the randomness involved in tree building.

Attempting to apply this U-statistic strategy to boosting, we managed to implement Boulevard with subsampling to create a kernel form. However, the actual difficulty comes from the serial dependence between boosted trees. There are two immediate solutions: the first is to verify that the covariance between component trees is of a lower magnitude than

of the variability caused by the error term conditioned on the given covariates, while the second is to verify that the gap between any Boulevard result and a proper U-statistic converges to zero at a rate faster than the U-statistic variance. Unfortunately, both the greedy tree building algorithm and the completely randomized tree strategy can complicate the relations between two particular trees, especially when their indices are far apart. Therefore though empirical studies yield good results, neither of the two immediate solutions is easy to justify theoretically.

Despite the existence of some up-to-date U-statistic research (Han and Qian, 2016), we still cannot take an easy approach to squeeze a covariance term into the U-statistic kernel. This is the reason why we choose to bruteforce the asymptotic distribution in our analysis of Boulevard.

5.2 Stochastic Contraction, Shrinkage, Dropout and Second Order Method

In terms of the ordinary boosting framework, one characteristic pattern of boosting iterations is that the signal is not uniformly distributed in time. The first few base learners, or trees in the context of this thesis, tend to be exposed to most of the signal, whereas the rest of the ensemble fits on small remainder terms or even random fluctuation if there is stationarity. This behavior brings two issue up. Empirically, tree boosting is dominated by the first few trees (as mentioned, over-specification), which complicates the training when

we involve stochastic strategies. Even with the same training data, the actual training paths and results may be substantially different once two boosting iterations do not agree at the beginning. Together with the volatility of decision trees, we can imagine the circumstance when the starting trees are by chance inaccurate, leading to the necessity of more trailing trees to correct. Theoretically, the decay of signal strength justifies the exercise of early stopping, requiring certain early stopping rate and invalidating any analysis that assumes we can build the ensemble to infinite size. However, the infinite ensemble is a better object to study its limiting distribution in the presence of either convergence or stationarity.

Compared to this ordinary framework, the shrinkage used in Boulevard results in an averaged ensemble. From the perspective of signal distribution, all trees are guaranteed to be exposed to certain signal level during training because of the shrinkage of training history. This effect, diminishing the influence of the starting trees, provides a means to balance the ensemble.

Instead of deterministically shrinking, another practice to adaptively weight the ensemble is through dropout. At each training iteration, some trees are randomly dropped out of the ensemble before we calculate the gradient, after which these trees are added back to the ensemble with smaller weights. While dropout is shown empirically to have improved the performance of boosted trees, a balanced dropout should as well produce an equally weighted ensemble, therefore can be viewed as a stochastic version of Boulevard. It is worth noticing that both Boulevard and dropout involve shrinkage that creates a contraction leading to potential convergence.

However, the side effect of introducing stochastic contraction is that the fixed point

cannot achieve consistency: the contraction on the training data should end somewhere strictly between the starting guess, which is 0 most of the time, and the observations Y, which is the target, preventing the fixed point from landing on the full signal in Y. As shown above, for Boulevard with L^2 loss we can rescale the prediction to compensate for this effect. However, the same strategy does not apply to general cases.

As suggested by original boosting implementation, the ideal learning rate λ^* should be decided by the second derivative in a Newton-Raphson style approximation to the root of the first order condition. Looking at Algorithm. 3.1 and using a generic loss function l, the optimal update to make at any point x_i should be

$$z_i \triangleq -\frac{\partial l(u_i, y_i)}{\partial u_i} \left(\frac{\partial^2 l(u_i, y_i)}{\partial u_i^2} \right)^{-1} \Big|_{u_i = \hat{f}_b(x_i)}.$$

When l is square loss whose second order is constant 1, this calculation reduces to Algorithm. 3.1. In practice, the (inverse of) second order term is sometimes omitted for both gradient descent and gradient boosting since the constant learning rate yields similar performance while preventing both the computation of the second order and the tendency of converging to a saddle point. However, this second order update is a better value for quantifying the signal level in the residuals.

This observation in particular brings an issue to any method involving shrinkage. As long as shrinkage reduces the signal level in the ensemble, in order for the stochastic contraction to land on a meaningful fixed point, we need to bridge between the fixed point and the actual signal. Their relation is decided by the learning rate that controls the location of the fixed point, and the curvature (second order) structure of the loss function

which controls the signal level of the fixed point. Any study attempting to justify the asymptotic behavior of such processes should provide a clear relationship between these two key factors.

5.3 Partially Linear Model Inference

Tree boosted VCM is not the only way to integrate tree boosting and partially linear models. One can easily fit a semi-parametric partially linear model by first fitting a linear model using the predictive covariates, then fitting ordinary boosted trees on the residuals. However, without the presence of proper regulations, it is hard to evaluate both the finite sample and the asymptotic behavior of this nonparametric tree ensemble, preventing us from performing inference with respect to the parametric part of the model. Moreover, after building the tree ensemble, we have no guarantee that the new residuals will be orthogonal to the linear part, creating a potential need for backfitting.

One solution to this concern is by interleaving the linear model and the nonparametric model with a boosting framework capable of providing distributional conclusions. Boulevard is a natural choice. We have the following algorithm for partially linear models with sample points as tuples of $(x_i, z_i, y_i), i = 1, \ldots, n$.

Algorithm 5.1 (Partially Linear Regression with Boulevard)**.**

- *Start with an initial nonparametric estimate* $\hat{f}_0 = 0$.

- *For a given current estimate $\hat{f}_b$, determine linear model coefficients $\hat{\beta}_b$ on $\left(x_i, y_i - \hat{f}_b(z_i)\right), i = 1, \ldots, n$ using least square.*
- *Calculate the gradient for the nonparametric part using residuals $y_i - \hat{\beta}_b^T x_i - \hat{f}_b(z_i)$,*

$$\delta_i \triangleq -\frac{\partial}{\partial u_i} \sum_{i=1}^{n} \frac{1}{2} (u_i - y_i)^2 \Big|_{u_i = \hat{\beta}_b^T x_i + \hat{f}_b(z_i)} = y_i - \hat{\beta}_b^T x_i - \hat{f}_b(z_i); \tag{5.1}$$

- *Generate a subsample $w \subset \{1, 2, \ldots, n\}$.*
- *Construct a tree regressor $t_b(\cdot)$ on $\{(x_i, \delta_i), i \in w\}$.*
- *Update the nonparametric part by learning rate $1 > \lambda > 0$,*

$$\hat{f}_{b+1} = \frac{b-1}{b} \hat{f}_b + \frac{\lambda}{b} t_b = \frac{\lambda}{b} \sum_{i=1}^{b} t_i.$$

We can show that, by writing $\mathbf{K}$ the corresponding tree structure matrix and H the hat matrix for the linear model, the nonparametric part of the model is estimated by

$$\hat{f}(Z) = \mathbf{K}(I - H)\left(\frac{I}{\lambda} + \mathbf{K}(I - H)\right)^{-1} Y,$$

whose form is similar to Boulevard. One future direction is to make analogous analysis to show its asymptotic behavior. In particular, this asymptotic analysis can lead to the inferential framework for the linear coefficients.

5.4 Varying Coefficient Models, Functional Trees and Tree Distillation

Functional trees in practice have good interpretability due to their clear covariate space segmentation. As mentioned above, functional trees can be treated as a special case of tree

boosted varying coefficient models when the varying coefficient mappings are piecewise constants showing the common linear coefficients in a flat region.

While standard functional trees inherit the building algorithm directly from CART by greedily evaluating the impurity reduction using submodels, our varying coefficient model and decision tree distillation provide an alternative that does not involve the construction and evaluation of numerous submodels.

Algorithm 5.2 (Functional Trees through Tree Boosted VCM)**.**

- *Start with sample* $(x_1, y_1), \ldots, (x_n, y_n), i = 1, \ldots, n.$
- *Duplicate x to create action covariates* $z_i = x_i$.
- *Construct a tree boosted varying coefficient model with varying coefficient mappings* $\hat{\beta}$ *as tree ensembles.*
- *Distill ensemble* $\hat{\beta}$ *to single trees* β^*.
- *Return the functional tree as* $\hat{y} = g(x^T\beta^*(x))$ *with g the link function.*

One possible future direction is to theoretically and empirically justify this method compared to the performance of standard functional tree construction.

5.5 Model Extrapolation and Manipulation

Different from the ordinary learning scheme, there are more and more circumstances nowadays where the purpose of fitting a predictive model is no longer to study the un-

derlying relationship between covariates and responses, but as a means to assign every covariate (subject) a score. One example is learning to rank, also referred to as information retrieval. It is a supervised learning problem with the input being a query and a set of subjects and the response being the ranks among the subjects indicating how well they match the properties of the given query. Imagine an online vendor selling apparels to customers. The query can be a generic description of what a customer wants, and the response should be a list of apparels matching their intent.

The common practice for learning to rank now is done through relevance scoring, which assigns a score for each query subject pair assessing how well they match after projecting them onto certain covariate space. The final ranking list is produced in the descending order of the relevance scores. In contrast to standard statistical inference discussing the behavior of the model with a new input, people may also be interested in knowing the feasibility of manipulating the model and the possible consequences afterwards. For example, this online vendor may decide to give higher scores to older products for clearance purposes and wonder, first, how to achieve so with their current learning model, and second, what outcomes they should expect.

This circumstance adds another dimension to our current understanding of model interpretability. Conceptually, we can name it *model extrapolation*, representing the feasibility and the expectation of perturbing the model itself once learned. Another motivation behind tree boosted varying coefficient models is due to this new aspect, as the linear model is easy to perturb by directly changing the coefficients, and is easy to analyze its outcome. Beside the vendor example we mentioned above showing the need for manually modifying

the model, we can in addition consider the following scenarios.

- Model monotonicity. From the point of view of model fairness we may intend to assure a monotonic relationship between certain covariate and the response. For instance, it is reasonable to expect that one is more likely to get a loan approved if they have a higher credit score, given that everything else on their profile stays the same. For tree boosted VCM, this relationship is described by the sign of the coefficient. Should *post hoc* adjustment be necessary, we can simply modify the corresponding coefficient to guarantee monotonicity.
- Expansion of the support of covariates. It may also be referred to as *warm start*, meaning that we want to improve an existing predictive model when new combinations of covariates emerge, in contrast to *cold start* for which we recollect training data and retrain the model when the covariate distribution changes. When the appearance of new covariate values takes place in the action space, tree boosted VCM can extrapolate the corresponding local parametric relationship using the existing model and certain similarity measure.

Further research can be done to expand these ideas based on suitable practical real world questions.

BIBLIOGRAPHY

Almudevar, A. A stochastic contraction mapping theorem. Unpublished Manuscript.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 46(3), 175–185.

Banerjee, M., I. W. McKeague, et al. (2007). Confidence sets for split points in decision trees. The Annals of Statistics 35(2), 543–574.

Basu, S., K. Kumbier, J. B. Brown, and B. Yu (2018). Iterative random forests to discover predictive and stable high-order interactions. Proceedings of the National Academy of Sciences, 201711236.

Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society. Series B (Methodological), 289–300.

Berger, M., G. Tutz, and M. Schmid (2017). Tree-structured modelling of varying coefficients. Statistics and Computing, 1–13.

Biau, G. (2012). Analysis of a random forests model. Journal of Machine Learning Research 13(Apr), 1063–1095.

Biau, G., L. Devroye, and G. Lugosi (2008). Consistency of random forests and other averaging classifiers. The Journal of Machine Learning Research 9, 2015–2033.

Blanchard, G., C. Schäfer, and Y. Rozenholc (2004). Oracle bounds and exact algorithm

for dyadic classification trees. In International Conference on Computational Learning Theory, pp. 378–392. Springer.

Blanchard, G., C. Schäfer, Y. Rozenholc, and K.-R. Müller (2007). Optimal dyadic decision trees. Machine Learning 66(2-3), 209–241.

Breiman, L. (1996). Bagging predictors. Machine learning 24(2), 123–140.

Breiman, L. (2001). Random forests. Machine learning 45(1), 5–32.

Breiman, L. et al. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statistical Science 16(3), 199–231.

Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen (1984). Classification and regression trees. CRC press.

Buergin, R. A. and G. Ritschard (2017). Coefficient-wise tree-based varying coefficient regression with vcrpart. Journal of Statistical Software 80(6), 1–33.

Bühlmann, P. and T. Hothorn (2007). Boosting algorithms: Regularization, prediction and model fitting. Statistical Science, 477–505.

Bühlmann, P. and B. Yu (2003). Boosting with the l 2 loss: regression and classification. Journal of the American Statistical Association 98(462), 324–339.

Bühlmann, P., B. Yu, et al. (2002). Analyzing bagging. The Annals of Statistics 30(4), 927–961.

Bühlmann, P. L. (2002). Consistency for lboosting and matching pursuit with trees and tree-type basis functions. In Research report/Seminar für Statistik, Eidgenössische

Technische Hochschule (ETH), Volume 109. Seminar für Statistik, Eidgenössische Technische Hochschule (ETH).

Candanedo, L. M. and V. Feldheim (2016). Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models. Energy and Buildings 112, 28–39.

Caruana, R. and A. Niculescu-Mizil (2006). An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning, pp. 161–168. ACM.

Chan, K.-Y. and W.-Y. Loh (2004). Lotus: An algorithm for building accurate and comprehensible logistic regression trees. Journal of Computational and Graphical Statistics 13(4), 826–852.

Chipman, H. A., E. I. George, and R. E. McCulloch (2010, 03). Bart: Bayesian additive regression trees. Ann. Appl. Stat. 4(1), 266–298.

Chipman, H. A., E. I. George, R. E. McCulloch, and T. S. Shively (2016). High-dimensional nonparametric monotone function estimation using bart. arXiv preprint arXiv:1612.01619.

Davies, A. and Z. Ghahramani (2014). The random forest kernel and other kernels for big data from random partitions. arXiv preprint arXiv:1402.4293.

Dheeru, D. and E. Karra Taniskidou (2017). UCI machine learning repository.

Domingos, P. (1997). Knowledge acquisition form examples vis multiple models. In Proceedings of the Fourteenth International Conference on Machine Learning, pp. 98–106. Morgan Kaufmann Publishers Inc.

Dunnett, C. W. (1955). A multiple comparison procedure for comparing several treatments with a control. Journal of the American Statistical Association 50(272), 1096–1121.

Fan, J., T. Huang, et al. (2005). Profile likelihood inferences on semiparametric varying-coefficient partially linear models. Bernoulli 11(6), 1031–1057.

Fan, J., W. Zhang, et al. (1999). Statistical estimation in varying coefficient models. The annals of Statistics 27(5), 1491–1518.

Fanaee-T, H. and J. Gama (2014). Event labeling combining ensemble detectors and background knowledge. Progress in Artificial Intelligence 2(2-3), 113–127.

Fernandes, K., P. Vinagre, and P. Cortez (2015). A proactive intelligent decision support system for predicting the popularity of online news. In Portuguese Conference on Artificial Intelligence, pp. 535–546. Springer.

Freund, Y., R. Schapire, and N. Abe (1999). A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence 14(771-780), 1612.

Freund, Y. and R. E. Schapire (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. In European conference on computational learning theory, pp. 23–37. Springer.

Friedberg, R., J. Tibshirani, S. Athey, and S. Wager (2018). Local linear forests. arXiv preprint arXiv:1807.11408.

Friedman, J., T. Hastie, and R. Tibshirani (2001). The elements of statistical learning, Volume 1. Springer series in statistics Springer, Berlin.

Friedman, J., T. Hastie, R. Tibshirani, et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics 28(2), 337–407.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting. Computational Statistics & Data Analysis 38(4), 367–378.

Gama, J. (2004). Functional trees. Machine Learning 55(3), 219–250.

Gibbons, R. D., G. Hooker, M. D. Finkelman, D. J. Weiss, P. A. Pilkonis, E. Frank, T. Moore, and D. J. Kupfer (2013). The computerized adaptive diagnostic test for major depressive disorder (cad-mdd): a screening tool for depression. The Journal of clinical psychiatry 74(7), 1–478.

Gordon, L. and R. A. Olshen (1984). Almost surely consistent nonparametric regression from recursive partitioning schemes. Journal of Multivariate Analysis 15(2), 147–163.

Han, F. and T. Qian (2016). Asymptotics for asymmetric weighted u-statistics: Central limit theorem and bootstrap under data heterogeneity. Under review at Annals of Statistics.(* alphabetical order of authorship).

Härdle, W., H. Liang, and J. Gao (2012). Partially linear models. Springer Science & Business Media.

Hastie, T. and R. Tibshirani (1993). Varying-coefficient models. Journal of the Royal Statistical Society. Series B (Methodological), 757–796.

He, H., J. Eisner, and H. Daume (2012). Imitation learning by coaching. In Advances in Neural Information Processing Systems, pp. 3149–3157.

Hochberg, Y. and Y. Benjamini (1990). More powerful procedures for multiple significance testing. Statistics in medicine 9(7), 811–818.

Hooker, G. (2007). Generalized functional anova diagnostics for high-dimensional functions of dependent variables. Journal of Computational and Graphical Statistics 16(3), 709–732.

Hothorn, T., P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner (2013). mboost: Model-based boosting, 2012. URL http://CRAN. R-project. org/package= mboost. R package version, 2–1.

Johansson, U. and L. Niklasson (2009). Evolving decision trees using oracle guides. In Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on, pp. 238–244. IEEE.

Johansson, U., C. Sönströd, and T. Löfström (2010). Oracle coached decision trees and lists. In International Symposium on Intelligent Data Analysis, pp. 67–78. Springer.

Johansson, U., C. Sönströd, and T. Löfström (2011). One tree to explain them all. In Evolutionary Computation (CEC), 2011 IEEE Congress on, pp. 1444–1451. IEEE.

Kaggle (2018). Housing price in beijing. `https://www.kaggle.com/ruiqurm/lianjia/home`.

Kaya, H., P. Tüfekci, and F. S. Gürgen (2012). Local and global learning methods for predicting power of a combined gas & steam turbine. In Proceedings of the International Conference on Emerging Trends in Computer and Electronics Engineering ICETCEE, pp. 13–18.

Liang, X., T. Zou, B. Guo, S. Li, H. Zhang, S. Zhang, H. Huang, and S. X. Chen (2015). Assessing beijing's pm2. 5 pollution: severity, weather impact, apec and winter heating. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 471(2182), 20150257.

Lichman, M. (2013). UCI machine learning repository.

Loh, W.-Y. and Y.-S. Shih (1997). Split selection methods for classification trees. Statistica sinica, 815–840.

Lou, Y., R. Caruana, and J. Gehrke (2012). Intelligible models for classification and regression. In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 150–158. ACM.

Lou, Y., R. Caruana, J. Gehrke, and G. Hooker (2013). Accurate intelligible models with pairwise interactions. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 623–631. ACM.

Lucas, D., R. Klein, J. Tannahill, D. Ivanova, S. Brandon, D. Domyancic, and Y. Zhang (2013). Failure analysis of parameter-induced simulation crashes in climate models. Geoscientific Model Development 6(4), 1157–1171.

Mallat, S. and Z. Zhang (1993). Matching pursuit with time-frequency dictionaries. Technical report, Courant Institute of Mathematical Sciences New York United States.

Mangasarian, O. L., W. N. Street, and W. H. Wolberg (1995). Breast cancer diagnosis and prognosis via linear programming. Operations Research 43(4), 570–577.

Melis, D. A. and T. Jaakkola (2018). Towards robust interpretability with self-explaining neural networks. In Advances in Neural Information Processing Systems, pp. 7786–7795.

Mentch, L. and G. Hooker (2014). Ensemble trees and clts: Statistical inference for supervised learning. arXiv preprint arXiv:1404.6473.

Mentch, L. and G. Hooker (2016). Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. The Journal of Machine Learning Research 17(1), 841–881.

Mentch, L. and G. Hooker (2017). Formal hypothesis tests for additive structure in random forests. Journal of Computational and Graphical Statistics 26(3), 589–597.

Moro, S., P. Cortez, and P. Rita (2014). A data-driven approach to predict the success of bank telemarketing. Decision Support Systems 62, 22–31.

Park, B. U., E. Mammen, Y. K. Lee, and E. R. Lee (2015). Varying coefficient regression models: a review and new developments. International Statistical Review 83(1), 36–64.

Polyak, B. T. and A. B. Juditsky (1992). Acceleration of stochastic approximation by averaging. SIAM Journal on Control and Optimization 30(4), 838–855.

Quinlan, J. R. (1987). Generating production rules from decision trees. In IJCAI, Volume 87, pp. 304–307. Citeseer.

Quinlan, J. R. (2014). C4. 5: programs for machine learning. Elsevier.

Rashmi, K. and R. Gilad-Bachrach (2015). Dart: Dropouts meet multiple additive regression trees. arXiv preprint arXiv:1505.01866.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016). Why should i trust you?: Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1135–1144. ACM.

Robbins, H. and S. Monro (1951). A stochastic approximation method. The annals of mathematical statistics, 400–407.

Rogozhnikov, A. and T. Likhomanenko (2017). Infiniteboost: building infinite ensembles with gradient descent. arXiv preprint arXiv:1706.01109.

Ruppert, D. (1988). Efficient estimations from a slowly convergent robbins-monro process. Technical report, Cornell University Operations Research and Industrial Engineering.

Scornet, E. (2016). Random forests and kernel methods. IEEE Transactions on Information Theory 62(3), 1485–1500.

Scornet, E., G. Biau, J.-P. Vert, et al. (2015). Consistency of random forests. The Annals of Statistics 43(4), 1716–1741.

Sorokina, D., R. Caruana, M. Riedewald, and D. Fink (2008). Detecting statistical interactions with additive groves of trees. In Proceedings of the 25th international conference on Machine learning, pp. 1000–1007. ACM.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1), 1929–1958.

Stone, C. J. (1977). Consistent nonparametric regression. The annals of statistics, 595–620.

Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. The annals of statistics, 1040–1053.

Tan, S., R. Caruana, G. Hooker, and Y. Lou (2017). Distill-and-compare: Auditing black-box models using transparent model distillation.

Tsanas, A. and A. Xifara (2012). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. Energy and Buildings 49, 560–567.

Tüfekci, P. (2014). Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. International Journal of Electrical Power & Energy Systems 60, 126–140.

Van der Vaart, A. W. (2000). Asymptotic statistics, Volume 3. Cambridge university press.

van der Vaart, A. W. and J. A. Wellner (1996). Weak convergence and empirical processes with applications to statistics. Springer.

Wager, S. and S. Athey (2017). Estimation and inference of heterogeneous treatment effects using random forests. Journal of the American Statistical Association (just-accepted).

Wager, S., T. Hastie, and B. Efron (2014). Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. The Journal of Machine Learning Research 15(1), 1625–1651.

Wager, S. and G. Walther (2015). Adaptive concentration of regression trees, with application to random forests. arXiv preprint arXiv:1503.06388.

Wager, S., S. Wang, and P. S. Liang (2013). Dropout training as adaptive regularization. In Advances in neural information processing systems, pp. 351–359.

Wand, M. P. and M. C. Jones (1994). Kernel smoothing. Crc Press.

Wang, J. C. and T. Hastie (2014). Boosted varying-coefficient regression models for product demand prediction. Journal of Computational and Graphical Statistics 23(2), 361–382.

You, S., D. Ding, K. Canini, J. Pfeifer, and M. Gupta (2017). Deep lattice networks and partial monotonic functions. In Advances in Neural Information Processing Systems, pp. 2981–2989.

Zhang, Q.-s. and S.-C. Zhu (2018). Visual interpretability for deep learning: a survey. Frontiers of Information Technology & Electronic Engineering 19(1), 27–39.

Zhang, T., B. Yu, et al. (2005). Boosting with early stopping: Convergence and consistency. The Annals of Statistics 33(4), 1538–1579.